FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Caracterização Tridimensional de Objetos Dinâmicos para Operações com Robôs Móveis

João António Brito Pires

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Andry Maykol Gomes Pinto

Co-orientador: Aníbal Castilho Coimbra de Matos

25 de Junho de 2018

Resumo

O número de robôs autónomos, para aplicação nas mais diversas tarefas, tem aumentado. A sua utilização permite retirar do Ser Humano a responsabilidade de tarefas que poderiam pôr em risco a sua vida ou que simplesmente ocupam a sua existência. Para que seja possível a integração destes robôs no seio do Homem e das suas atividades, é necessário que estes estejam dotados com complexos sistemas de perceção, cuja principal função passa pela vigilância do ambiente e das interações que o sistema tem com este.

Desta forma, a presente dissertação descreve o desenvolvimento de um sistema multissensorial capaz de detetar e caracterizar objetos no ambiente circundante. Foi criado combinando a informação de um sistema baseado em *Convolutional Neural Networks*, que realiza a deteção e classificação de objetos em imagens, e a informação tridimensional recolhida de uma *pointcloud*. Com a associação destes dados, o sistema é capaz de identificar os objetos existentes, a classe a que pertencem e qual a sua localização e dimensões, indicando também, se estes se encontrarem em movimento, a sua velocidade e direção.

A partir das experiências concretizadas, verificou-se que o sistema desenvolvido representa uma mais valia para os módulos de perceção de sistemas autónomos móveis, uma vez que disponibiliza informação útil para aplicações de patrulhamento e vigilância, assim como informação vital para sistemas de navegação.

Abstract

The number of autonomous robots, for application in the most diverse tasks, has increased. Their use allows them to take responsibility for life-threatening tasks of the Human being or that simply take over their lives. In order to be able to integrate these robots into the midst of Man and its activities, it is necessary that they be equipped with complex systems of perception, whose main function is the surveillance of the environment and the interactions that the system has with it.

Thus, the present dissertation describes the development of a multisensory system capable of detecting and characterizing objects in the surrounding environment. It was created by combining information from a system based on Convolutional Neural Networks, which performs the detection and classification of objects into images, and gathered three-dimensional information from a pointcloud. With the association of these data, the system is able to identify the existing objects, the class to which they belong and their location and dimensions, also indicating, if they are in motion, their speed and direction.

From the experiments carried out, it was verified that the developed system represents an added value for the autonomous mobile systems perception modules, since it provides useful information for patrolling and surveillance applications, as well as vital information for navigation systems.

Agradecimentos

Em primeiro lugar, agradeço ao meu orientador, Professor Doutor Andry Pinto, por toda a paciência, disponibilidade e dedicação demonstrada ao longo desta dura etapa. A sua exigência e integridade serão levadas comigo para a vida profissional que se avizinha.

A toda a equipa do CRAS, mas em especial à Rita e à Alexandra, por todos os conselhos e ajuda dada, mesmo que fora de horas, a vocês o meu muito obrigado!

À minha família, por todos os sacrifícios feitos por mim, por todos os ensinamentos que nenhuma escola me poderia dar. Aos meus pais, à minha irmã, mas especialmente à minha avó, que me educou para que fosse o Homem que hoje sou.

À Sofia, por todo o carinho e motivação dados ao longo destes anos, por ser o meu pilar nos momentos mais conturbados.

Por fim, mas não menos importantes, ao Miguel, ao Rui e à Beatriz, por todo o apoio, por todos os risos, por todos os momentos passados. Mais do que grandes colegas, são grandes amigos.

A todos eles, e tantos outros que marcaram o meu percurso,

Muito Obrigado.

João Pires

"Great men are not born great, they grow great."

Mario Puzo,The Godfather

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
CNN	Convolutional Neural Network
CRAS	Centre of Robotics and Autonomous Systems
FPS	Frames Per Second
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
mAP	mean Average Precision
RaDAR	Radio Detection And Ranging
RNN	Recurrent Neural Network
SoNAR	Sound Navigation And Ranging

Capítulo 1

Introdução

1.1 Contexto

O CRAS, local no qual foi desenvolvida a presente dissertação, é um centro de investigação que une esforços na área de sistemas de navegação autónomos, com ênfase em sistemas aquáticos. O aumento da capacidade de processamento e memória dos computadores, aliado à redução das suas proporções, proporciona a exploração e desenvolvimento de novas aplicações, cada vez mais complexas, para aplicação nestes sistemas.

Atualmente, os veículos autónomos de superfície (ASV) são já utilizados para fins científicos, militares e comerciais. São empregues na vigilância da costa, inspeção de estruturas e até mesmo na recolha de informações como a qualidade e temperatura da água. Contudo, para realizarem as suas missões de forma autónoma, têm de ser capazes de superar alguns entraves à sua navegação. Estes devem estar dotados de sistemas habilitados a detetar e compreender o ambiente que os rodeiam, de forma a poderem-se deslocar sem o perigo de colisões. Além disso, a identificação e caracterização dos objetos permite o acompanhamento dos mesmos, a previsão dos seus próximos estados e quais os seus movimentos.

1.2 Objetivos

A presente dissertação pretende desenvolver um sistema computacional para aplicação em robôs móveis, capaz de modelar o cenário de operações e as movimentações ocorridas neste. Deve identificar objetos à sua volta, indicando a sua posição relativa e dimensões aproximadas, a três dimensões. Será dada ênfase aos objetos dinâmicos, sendo calculada também a sua velocidade e direção.

Assim, esta dissertação apresenta como principais objetivos:

- Estudo e avaliação de soluções para identificação de objetos;
- Estudo de métodos para determinação das características tridimensionais dos objetos (posição, velocidade, direção e dimensão);

- Implementação de algoritmos para obtenção das características;
- Validação experimental com dados recolhidos em ambientes controlados.

1.3 Motivação

Os veículos autónomos de superfície são utilizados, entre outras, para aplicações de inspeção e vigilância. Para que as suas missões sejam concretizadas com sucesso, é necessário a existência de um módulo de perceção apto a estimar o meio em que se encontra e as movimentações ocorridas, fornecendo ao sistema ferramentas essenciais a tarefas como o desvio de obstáculos e *tracking*. Pode-se afirmar que este é um dos componentes chave destes sistemas, pelo que deve ser o mais eficaz, fiável e preciso possível. Contudo, esta tarefa tem-se mostrado complexa e computacionalmente dispendiosa.

Uma grande parte dos algoritmos existentes não permite o processamento dos dados em tempo real, o que invalida a possibilidade da aplicação em sistemas robóticos móveis, dos quais fazem parte os ASV. Isto deve-se quer à dimensão dos dados capturados, como à quantidade elevada de cálculos necessários para o seu processamento, que exige o consumo de muitos recursos computacionais.

Os tipos de sensores utilizados mais frequentemente para a aquisição dos dados são os sensores óticos. Apesar da grande quantidade de informação útil, as imagens recolhidas padecem de alguns problemas aquando da sua aquisição. Uma vez que o princípio físico da captura de uma imagem é dependente da luz, a qualidade da captura dependerá da quantidade existente no ambiente, pelo que os dados obtidos sofrerão com movimentações das embarcações ou condições climatéricas mais adversas. Para além disso, movimentos mais bruscos de mudança de direção, ou oscilação provocada pela ondulação marítima, podem originar distorções da imagem. Estas alterações trazem um maior grau de incerteza a métodos que, por si só, detêm algum erro associado, como a deteção de objetos recorrendo a histogramas de cor ou o cálculo de distâncias recorrendo a imagens estereocópicas.

Assim sendo, o desenvolvimento de um sistema multi-sensorial, que conjugue a informação de uma câmara e de um sensor medidor de distâncias LiDAR (*Light Detection and Ranging sensor*), proporcionaria uma caracterização tridimensional do espaço envolvente mais precisa e robusta. Desta forma, o sistema é dotado de ferramentas para uma navegação autónoma mais segura, além da possibilidade de fornecer uma série de informações úteis para as mais variadas aplicações.

1.4 Estrutura da Dissertação

Este documento encontra-se dividido em quatro capítulos. No presente capítulo é feita uma introdução ao tema, assim como é apresentada a motivação e os principais objetivos desta dissertação.

No capítulo 2 é apresentado o estudo da literatura realizado, no que diz respeito às técnicas e métodos desenvolvidos para a identificação dos objetos e das suas características inerentes.

No capítulo 3 são apresentados os métodos e algoritmos utilizados para a construção do sistema, sendo apresentadas algumas justificações para as escolhas efetuadas, através de factos teóricos e resultados de testes realizados. Ainda neste capítulo, são mostrados e comentados os resultados de todas as etapas implementadas, assim como o resultado final obtido.

Por fim, no capítulo 4 são apresentadas as conclusões do trabalho desenvolvido, assim como sugestões de propostas para trabalho futuro.

Capítulo 2

Estado da Arte

No presente capítulo são expostas teoricamente as principais temáticas presentes nesta dissertação. Para além de alguns conceitos introdutórios, nas secções que se seguem são apresentados trabalhos, disponíveis na literatura, que introduziram algumas soluções para as questões abordadas. Assim, na secção 2.1 é apresentada a temática da deteção e classificação de objetos, condição necessária para a posterior caracterização tridimensional, apresentada na secção 2.2. Esta última apresenta-se dividida em dois temas, a determinação da localização e dimensão dos objetos e, por fim, o cálculo da velocidade e direção destes.

2.1 Deteção e Classificação de Objetos

O aumento da capacidade de processamento dos computadores e a diminuição das suas dimensões, aliado à dedicação de uma grande comunidade científica empenhada na área, levou a um aumento no desenvolvimento de veículos autónomos. Entre outras, estes necessitam de ser dotados de ferramentas de perceção que permitam conhecimento do ambiente em redor, detetando obstáculos ou reconhecendo perigos associados à interação com outros sistemas móveis, como o caso da existência de rotas de colisão. Inicialmente estes mecanismos são desencadeados pela deteção de objetos, que deve ser realizada da forma mais eficiente e precisa possível. Para tal, estes sistemas estão usualmente equipados com uma grande variedade de sensores, destacando-se as câmaras e o LiDAR.

Quanto à deteção através de imagens recolhidas por câmaras, nas últimas décadas têm sido explorados métodos para o efeito, os quais passam pela aplicação de métodos de remoção do *background*, extração de características ou até de modelos estatísticos [7]. Com a aplicação destes sistemas em robôs móveis, foram surgindo dificuldades na utilização dos métodos desenvolvidos até então. Foi necessário passar a ter em conta a definição das imagens captadas em movimento, as variações de iluminação e a mudança da aparência e dimensão dos objetos com o movimento, pelo que todas estas alterações exigiram a adaptação do que tinha sido estudado até então. Em [8] é compilado num único documento os desenvolvimentos observados, apresentando os trabalhos *state-of-the-art* realizados nos últimos anos.

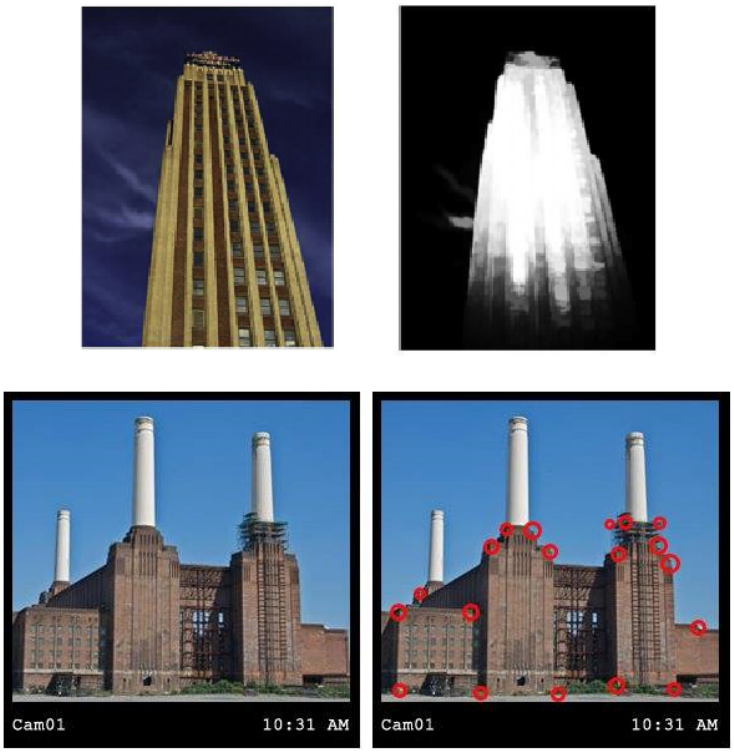

Figura 2.1: Métodos de deteção de objetos: imagem superior - remoção do fundo, adaptado de [1]; imagem inferior - deteção de cantos [2]

Quanto à utilização do LiDAR, esta prende-se com a representação tridimensional muita precisa e fiável obtida do ambiente envolvente, uma vez que é capaz de captar centenas de milhares de pontos por segundo. Em [3], por exemplo, os pontos captados são substituídos por uma grelha tridimensional, ocupada por cubos conhecidos como *voxels*, que são considerados como ocupados quando existem pontos dentro dos seus limites, permitindo a deteção de obstáculos mesmo quando estes são pequenos e com forma irregular. Já em [9] é usada a mesma grelha para detetar objetos em movimento, através da inconsistência das *voxels* ocupadas entre dois *scans* consecutivos. Para além disso, é apresentada uma forma de identificar o tipo de objeto detetado, de entre quatro classes: peão, bicicleta, carro e autocarro, recorrendo a redes neuronais para efetuar a classificação.

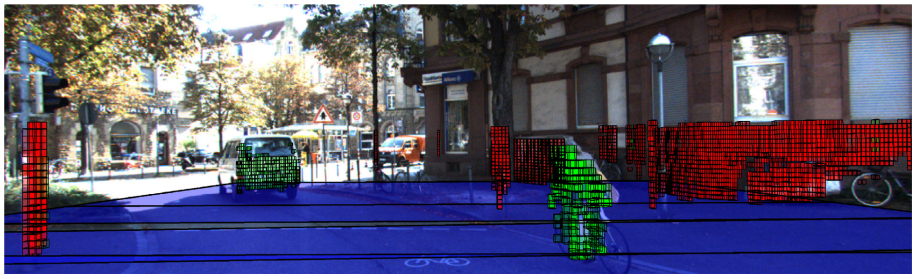

Figura 2.2: Grelha tridimensional (*voxel grid*) com objetos detetados [3]

A classificação dos objetos é uma das características que pode ser retirada no momento em que ocorre a deteção, sendo bastante útil para aplicações de vigilância, em que o sistema deve registar tudo o que aconteceu em seu redor. Nos últimos anos o *deep learning* tem alcançado resultados impressionantes nesta área, produzindo métodos mais precisos e eficientes que os restantes e conquistando, com isso, lugares cimeiros em competições, o que tem levado esta área do *Machine Learning* a ser alvo de um crescente interesse da comunidade científica [10]. Do ponto de vista da estrutura da rede neuronal utilizada, [10] divide os métodos existentes em três tipos: os baseados em *Convolutional Neural Networks* (CNN), os baseados em *Recurrent Neural Networks* (RNN) e os restantes. Os métodos CNN são os mais eficazes na extração de características e na classificação de imagens. Um excelente exemplo da sua aplicação é o trabalho apresentado por [11], o qual consegue detetar e classificar objetos recorrendo a uma única rede CNN, tudo isto em menos de 25 milissegundos. Já os métodos RNN, devido à sua natureza recursiva, são mais adequados para a modelação de sequências, sendo um exemplo da sua aplicação o sistema criado em [12], para o seguimento de objetos.

2.2 Caracterização Tridimensional

Após a sua deteção, a caracterização de um objeto no espaço tridimensional permite um conhecimento mais completo do ambiente. O conhecimento da sua localização exata relativamente ao sistema, assim como a sua dimensão, velocidade e direção, dotam o sistema de informações que permitem a previsão do futuro deslocamento e posição do objeto e, com isso, a possibilidade de evitar a colisão com os mesmos. Assim, nas próximas secções são abordadas formas, encontradas na literatura, de obter estas características.

2.2.1 Localização e Dimensão dos Objetos

Baseando-se no tipo de sinal de entrada, [13] classifica os métodos de localização de objetos em dois tipos: os diretos e os baseados em visão. Aos primeiros associam-se sensores como o radar, o sonar ou o LiDAR, que fornecem informação direta da distância graças à interpretação de sinais enviados e o seu *feedback*, seja na forma de ondas sonoras ou eletromagnéticas. Alcançam excelentes *performances*, sendo muito usados em aplicações para condução autónoma, devido aos seus resultados muito precisos e rápidos. O LiDAR, por exemplo, obtém uma nuvem de pontos densa o suficiente para delinear as formas dos objetos, sendo o agrupamento desses pontos e a deteção de linhas e arestas que permite a sua localização tridimensional e estimação de dimensões [14]. Um dos métodos possíveis para a estimação das dimensões é a utilização das grelhas tridimensionais, usadas em [3]. Uma vez que a dimensão de cada um desses espaços é conhecida, a contagem do número de espaços preenchidos vertical e horizontalmente permite estimar as dimensões dos objetos.

No caso dos métodos baseados em visão, são utilizadas câmaras como forma de captação de informação, onde a localização é baseada em conceitos de perceção visual humana. O uso destes sensores torna o sistema muito mais barato, comparativamente aos primeiros, contudo o seu

processamento é mais exigente e dispendioso. É encontrado na literatura uma extensa utilização de câmaras *stereo* para o efeito [15, 16, 17]. Estes sistemas de visão procuram simular a forma como o ser humano é capaz de ver objetos a três dimensões e perceber a profundidade a que se encontram, combinando a perspetiva de duas imagens. Aliado a estes sistemas está a aplicação do método de triangulação, que permite obter a localização de um objeto recorrendo à disparidade dos seus pontos entre duas imagens retiradas de câmaras diferentes, a uma dada distância, no mesmo momento.

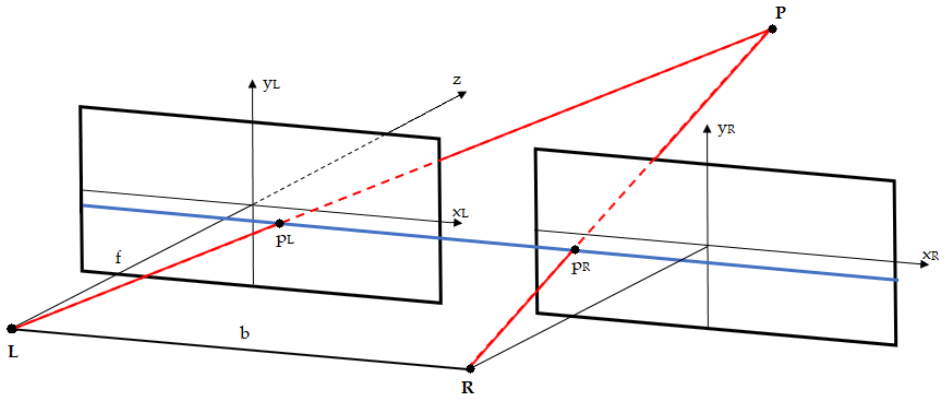

Figura 2.3: Representação tridimensional do método de triangulação

Para a aplicação deste método é necessário saber a distância focal das câmaras (f) e a distância entre elas (b). Assim, juntamente com a disparidade ($x_L - x_R$), obtém-se o valor da profundidade (Z) do objeto (2.1).

$$Z = \frac{b \times f}{x_L - x_R} \tag{2.1}$$

Após a obtenção da profundidade, utilizando a imagem da câmara esquerda como referência, as coordenadas X e Y podem ser obtidas, para completar a localização tridimensional, através das expressões 2.2 e 2.3, sendo (x_L, y_L) a localização do ponto projetado no plano da imagem esquerda (p_L).

$$X = \frac{x_L \times Z}{f} \tag{2.2}$$

$$Y = \frac{y_L \times Z}{f} \tag{2.3}$$

O método da triangulação pode ser usado também para a determinação das dimensões do objeto. Sabendo a profundidade a que se encontra, é possível fazer uma projeção das extremidades detetadas na imagem e, com isso, extrapolar as dimensões. Em [18] este método é adaptado, sendo criadas equações capazes de calcular as dimensões tendo em conta a localização do objeto na imagem e as dimensões da sua *boundary box*.

Apesar dos resultados deste método serem bastantes satisfatórios, com técnicas capazes de estimações com exatidão superior a 90% [16], têm sido efetuados estudos que adicionam a estas medições novas técnicas para garantir a exatidão dos dados. Em [19], por exemplo, são aplicadas tabelas com conjuntos de valores que relacionam a disparidade e a distância real do objeto, recolhidas através de enumeros ensaios, aos valores calculados da triangulação. Enquanto isso, em [17] são usadas redes neuronais para a otimização dos valores.

2.2.2 Velocidade e Direção

Tal como para a localização, a velocidade e direção podem ser obtidas de forma direta, com sensores como os radares ou o LiDAR, ou recorrendo à visão computacional. O radar é um dos sensores mais utilizados para a determinação de velocidade, sendo amplamente conhecida a sua aplicação por parte das forças policiais em ações de fiscalização, por exemplo. O funcionamento deste sensor baseia-se no Efeito de Doppler, que é nada mais do que a alteração da frequência de uma onda relativamente ao seu observador devido ao deslocamento da fonte emissora. Um dos exemplos mais facilmente reconhecidos é a mudança do tom na buzina de um carro que esteja em andamento. Nestes sensores é emitida uma onda eletromagnética, que é refletida pelo objeto mais próximo. A onda refletida, caso o objeto esteja em movimento, tem uma frequência diferente, pelo que sabendo essa diferença (frequência de Doppler) e a frequência da onda do radar, pelo teorema de Doppler é possível calcular a velocidade do objeto refletor [20].

$$v = \frac{f_d \times c}{2 \times f_{ef}} \tag{2.4}$$

- f_d - frequência de Doppler (Hz)
- f_{ef} - frequência do radar (Hz)
- c - velocidade da luz (m/s)

Quanto aos sistemas baseados em visão, um método comum é a análise de determinadas características, como linhas ou cantos, entre duas *frames* capturadas consecutivamente. Em [21] é usada a *boundary box* dos objetos detetados e o deslocamento do seu ponto central para calcular a velocidade em pixeis/s, sendo depois o valor convertido para Km/h tendo em conta a dimensão da estrada, entre linhas, e a relação que esta ocupa na imagem. As entraves deste método prendem-se com a obrigatoriedade da imobilização do sistema, já que é necessário dimensões de referência para se poder calcular o deslocamento. Já em [22] são utilizadas as dimensões reais de carros e as suas projeções em imagens para conseguir determinar o seu deslocamento entre *frames* consecutivos e, assim, determinar a velocidade. Um dos pontos negativos desta alternativa é que se pretendesse-se aplicar a outros objetos, seria necessário armazenar informações das suas dimensões reais e, no momento da aplicação, detetar qual o objeto a que era pretendido observar o movimento.

Contudo, estes métodos são computacionalmente dispendiosos, além de serem afetados por discrepâncias resultantes das diferentes dimensões captadas dos objetos, devido à alteração da sua

orientação com o movimento. Por essa razão, [23] apresenta um método baseado no fluxo ótico e que tem em conta não duas, mas três *frames* consecutivas para a determinação do contorno dos objetos, velocidade e direção. Já em [24] é apresentada uma solução oposta, sendo determinada a velocidade e direção de múltiplos objetos utilizando apenas uma imagem e os borrões captados do seu deslocamento.

Capítulo 3

Caracterização Tridimensional de Objetos

O presente capítulo pretende expor um método para caracterizar tridimensionalmente objetos e apresentar os resultados obtidos. Assim, este encontra-se organizado em 4 secções: é apresentado o problema alvo de estudo e identificados os tópicos de discussão (Secção 3.1), indicadas as tecnologias e ferramentas utilizadas (Secção 3.2), os métodos propostos para implementação (Secção 3.3) e, por fim, os resultados da aplicação (Secção 3.4).

3.1 Problema e Formulação Teórica

Para que seja possível a existência de veículos ou outros sistemas robóticos completamente autónomos, é necessário que sejam incorporadas funcionalidades avançadas de perceção que forneçam dados do ambiente circundante. Desta forma é possível dotá-los da capacidade de desvio de obstáculos, seguimento de determinados objetos ou identificação de possíveis rotas de colisão. Por essa razão, os sistemas de perceção são há muitos anos alvo de estudo.

De forma a criar estas aplicações, um conjunto de questões têm de ser solucionadas. Que características do ambiente são necessárias obter e interpretar para o desenvolvimento de funcionalidades autónomas que visem aumentar a segurança da circulação dos veículos no conjunto diversificado de cenários e condições ambientais (urbanos, não urbanos, etc)? Que estratégias de perceção (e respetiva complementaridade sensorial) é que poderão ser utilizadas para obter o conjunto de informação considerada crítica para a segurança da circulação rodoviária? Que requisitos computacionais serão imprescindíveis para se obter a performance em tempo-real necessária para tornar a condução autónoma uma realidade? A presente dissertação pretende estudar e implementar uma aplicação que tenha em conta estas questões.

Assim, e após a apresentação das tecnologias e ferramentas utilizadas para o sistema de perceção, é abordada a temática da deteção visual de objetos recorrendo a métodos de aprendizagem computacional existentes na literatura científica, com foco no *deep learning*, e o acompanhamento temporal e extração das características de movimento dos objetos que se situem à frente do veículo.

3.2 Tecnologias e Ferramentas do Sistema de Perceção

A representação do ambiente circundante do veículo autónomo deve ser precisa e fiável, para que seja possível a sua circulação num conjunto de cenários e condições ambientais diversificadas de forma segura. Para tal, foram acauteladas três medidas que pretendem assegurar este objetivo: a utilização de múltiplos sensores para recolha e cruzamento de dados, um *software* desenvolvido especialmente para a área da robótica e a utilização de bibliotecas vulgarmente aplicadas nesta área, otimizados para o processamento dos dados recolhidos. Os desenvolvimentos ocorridos nesta tese tiveram por base um sistema de perceção com as características e estrutura apresentadas de seguida.

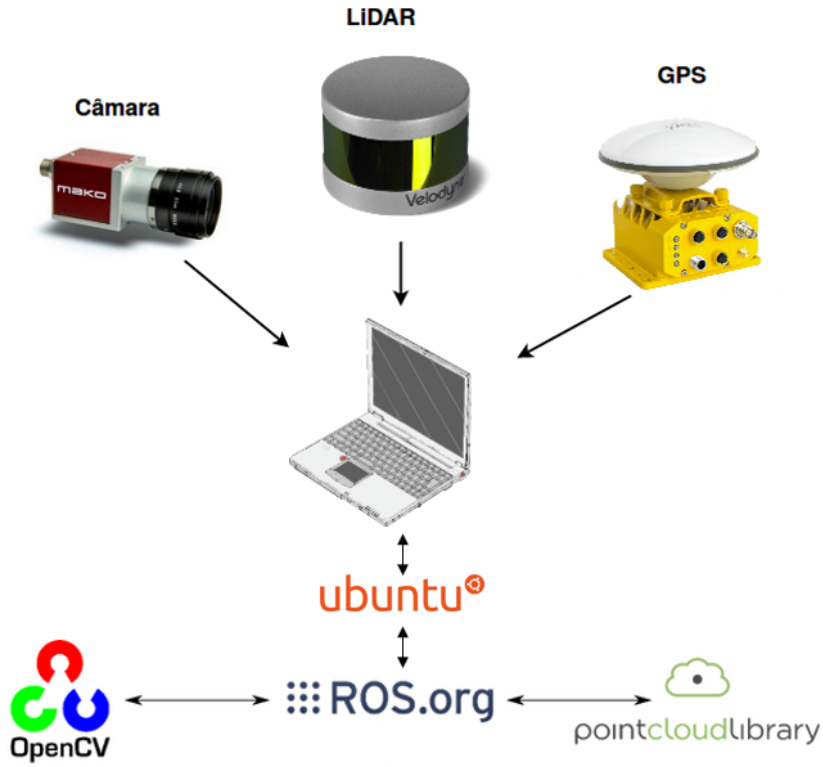

Figura 3.1: Arquitetura do sistema utilizado

Ao nível do hardware, a escolha dos sensores recaiu num sistema já existente no CRAS, criado para outros projetos com AUVs e ASVs, o que permitiu uma adaptação e integração rápida no sistema em desenvolvimento. É constituído por uma câmara (Mako G-125), um LiDAR (Velodyne PUCK VLP-16) e um sistema de posicionamento global (SwiftNAVIGATION RTKGPS). Com estes, conseguimos obter até 30 imagens e 300.000 pontos por segundo, assim como a localização relativa a uma estação base, com apenas alguns centímetros de erro.

Este sistema está implementado em ROS, sendo a linguagem de programação escolhida o C++. O ROS é um *software open-source* concebido com o objetivo de simplificar o processo de desenvolvimento de aplicações na área da robótica. Constituído por um conjunto de bibliotecas e

ferramentas, oferece uma plataforma capaz de gerir processos e respetiva comunicação entre estes. É assente num conceito de comunicação denominado *Publisher-Subscriber*, em que processos (chamados de Nós no contexto deste *software*) produzem informação (*Publishers*) que é publicada em Tópicos (canais de transmissão das mensagens). Os nós que pretendam receber essas mensagens subscrevem o tópico (*Subscribers*), sendo-lhes depois enviadas as mensagens à cadência a que são publicadas. Esta plataforma permite assim a construção de sistemas complexos, que integrem um vasto leque de sensores e atuadores, mesmo que de diferentes fornecedores, reduzindo preocupações de incompatibilidade de produtos e dados.

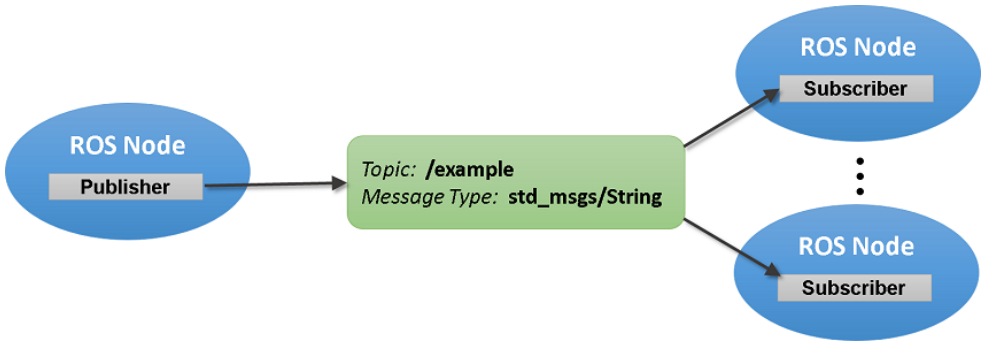

Figura 3.2: Comunicação de grupo *Publisher-Subscriber* [4]

Para o processamento dos dados, foram usadas duas bibliotecas amplamente utilizadas na área das aplicações de percepção e visão computacional e que, para além disso, possuem interfaces com o ROS.

O OpenCV é aplicado no processamento de imagens, em aplicações de visão computacional e *machine learning*. Tem suporte para linguagens como C/C++, Python ou Java, e funciona em diversos sistemas operativos. Contém mais de 2500 algoritmos otimizados para tarefas que permitem, por exemplo, a deteção e reconhecimento de faces ou a identificação e seguimento de objetos. Permite alcançar os resultados desejados cumprindo as restrições temporais dos sistemas de tempo real, tirando proveito de funcionalidades implementadas na biblioteca para usufruir das capacidades da aceleração do GPU do computador e do processamento multi-core.

O PCL surgiu da colaboração de dezenas de empresas e universidades, espalhadas pelo mundo, no desenvolvimento de uma biblioteca para processamento de *pointclouds*. Contém algoritmos *state-of-the-art* para tarefas como a filtragem, segmentação ou o reconhecimento de objetos baseando-se na sua forma geométrica.

3.3 Métodos Propostos

Até chegar-se a um resultado final, no qual são apresentados ao utilizador os objetos encontrados e as suas características, são necessárias um conjunto de etapas de processamento de dados,

que vão desde a identificação dos objetos até ao calculo da sua posição, por exemplo. Seguidamente são representadas, recorrendo a um fluxograma, as etapas fundamentais do algoritmo concebido, para que depois, nas subsecções que se seguem, sejam descritos ao pormenor os métodos aplicados.

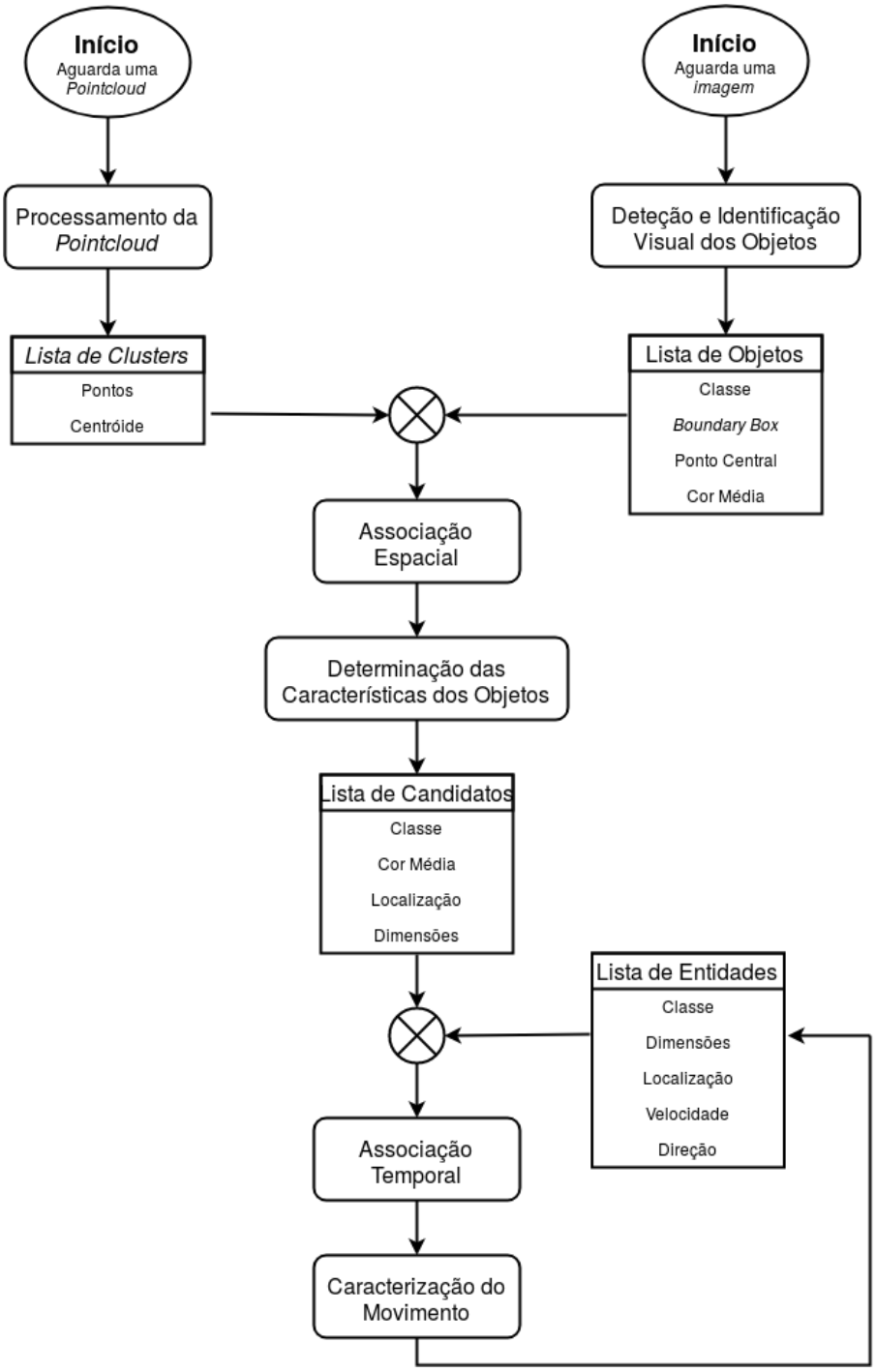

Figura 3.3: Fluxograma das etapas do algoritmo desenvolvido

3.3.1 Deteção e Classificação de Objetos

A deteção de um objeto é o primeiro passo no processo, uma vez que o sistema deve reconhecer a sua existência para poder desencadear o processamento da caracterização. Como referido no capítulo 2, existem muitas formas de realizar este procedimento, recorrendo a imagens ou *point-clouds*, usando histogramas de cor, identificação pela forma geométrica ou até mesmo *machine learning* e redes neuronais. Uma vez que esta área tem sido alvo de intenso estudo e o âmbito desta dissertação demarca-se um pouco do tópico de deteção de objetos, pretende-se apenas fazer uso dos seus resultados. Optou-se assim pelo estudo de métodos e projetos com provas dadas para aplicação, sendo estes escolhidos pelo seu estatuto de *state-of-the-art* na literatura. Assim, na Tabela 3.1 são apresentados os métodos comparados, as métricas de avaliação utilizadas e quais os resultados.

Modelo	**mAP-50**	**FPS**	**Artigo**	**Projeto**
SSD300	41.2	46	[25]	[26]
RetinaNet 101-800	57.5	5	[27]	[28]
FPN FRCN	59.1	6	[29]	[30]
YOLOv3-608	57.9	20	[31]	[32]
YOLOv3-Tiny	33.1	120	[31]	[32]

Tabela 3.1: Comparação da precisão e tempo de processamento de métodos de classificação de objetos

As métricas utilizadas para avaliar a *performance* destes algoritmos foram a precisão das classificações e a velocidade de processamento. A primeira é apresentada na forma mAP (*mean Average Precision*), na qual é medida a percentagem de previsões realizadas que apresentam classificações corretas e valor de IoU (*Intersection over Union*) elevado. Este IoU refere-se à interseção entre o local onde o objeto se encontra na imagem e a previsão do mesmo, como ilustrado na Figura 3.4. Todos os valores apresentados estão de acordo com a COCO mAP. Quer isto dizer que todos os algoritmos apresentados foram testados com o COCO *Dataset* [33] e que a classificação foi considerada como correta quando a taxa de IoU era superior a 50%.

Já a velocidade de processamento é indicada na forma de FPS (*Frames Per Second*), o que permite compreender de forma intuitiva se estes algoritmos conseguem cumprir com as restrições temporais inerentes a este tipo de sistemas, em que a classificação deverá ser realizada a um ritmo superior ao empregue pela captura de informação do sensor, ou seja, da câmara. Após a análise dos resultados da Tabela 3.1, conjugando precisão, velocidade de processamento e a existência de uma implementação em C, com capacidade de adaptação para ROS, a escolha recaiu no projeto YOLO - *You Only Look Once* [31].

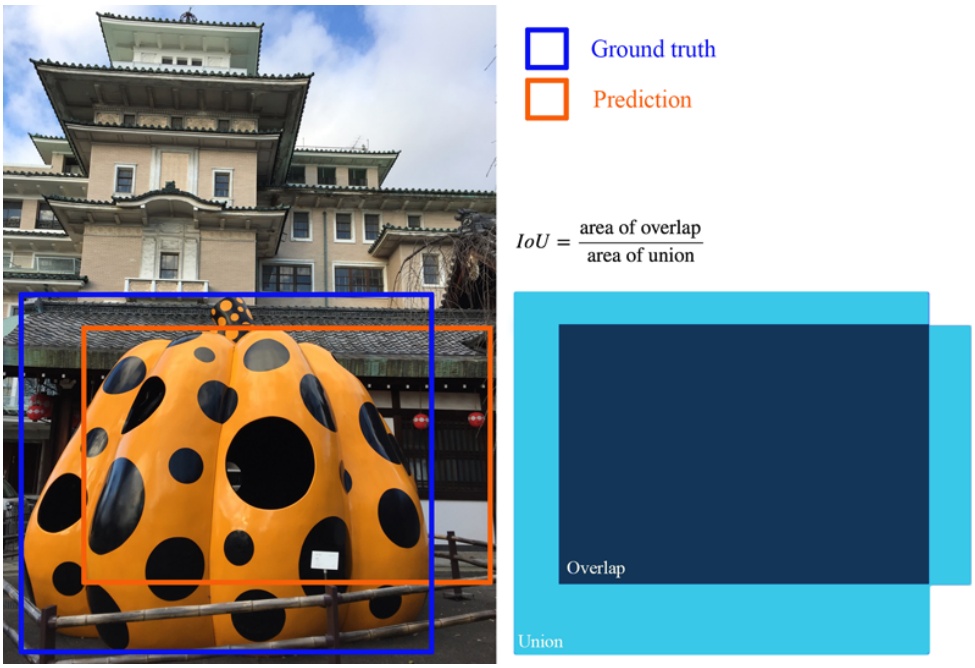

Figura 3.4: Exemplo ilustrativo da determinação do IoU [5]

Apresentado em 2015 e atualmente na sua terceira versão, intitula-se como um dos mais rápidos algoritmos de deteção e identificação de objetos, chegando a ser cem a mil vezes mais rápido que a sua competição direta. É capaz de localizar em imagens a posição e classe dos objetos presentes, apresentando estes resultados na forma de quadriláteros com cores distintas, apelidados de *boundary box*. O seu funcionamento é dependente de uma única rede neuronal convolucional, que prevê a existência de múltiplas *boundary box* ao longo da imagem e a probabilidade de efetivamente existirem objetos de determinada classe nessa localização, pelo que é extremamente rápido comparado com outros métodos, que utilizam centenas de redes neuronais. Aliado a essa característica distinta, o treino desta rede é feito com imagens integrais de extensas bases de dados e otimizado para a deteção de objetos, pelo que é reduzido o número de erros ocorridos e a lista de classes possíveis de detetar é extensa.

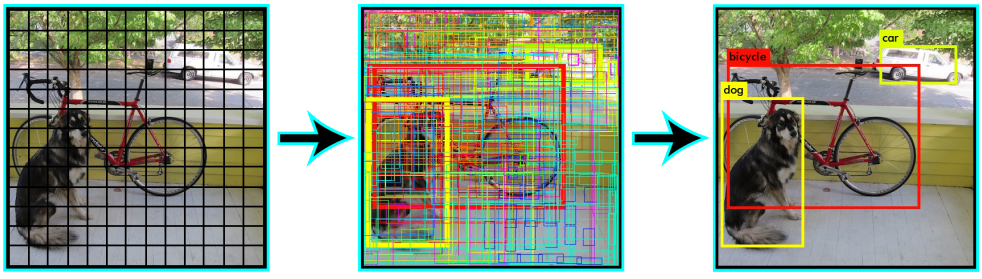

Figura 3.5: Etapas na localização e classificação dos objetos na imagem [6]

Com algumas alterações, graças a distribuições deste projeto adaptadas ao ROS, é possível a

sua utilização com os dados obtidos na câmara do sistema, conseguindo-se no fim desta operação um vetor de objetos encontrados na imagem.

Posteriormente, são extraídas um conjunto de características visuais que estão intimamente associadas a cada um dos objetos detetados por YOLO. Essas características são importantes para a definição e representação quantitativa de cada objeto detetado, e que compreendem: a classe, as coordenadas (u,v) dos quatro cantos e do ponto central da *boundary box* e a média da cor de todos os pixeis contidas nela.

Para além destas características de natureza visual, torna-se relevante obter um conjunto de características métricas, obtidas com recurso à informação disponível pelo LiDAR.

3.3.2 Processamento da *Pointcloud*

De forma a tornar a caracterização mais precisa, os dados obtidos pela câmara serão cruzados com representações tridimensionais do espaço envolvente, numa fusão sensorial. Esta representação tridimensional, correntemente chamada de *pointcloud*, agrega milhares de pontos por segundo, podendo ser usada para determinar a distância de objetos ao sensor com apenas alguns centímetros de erro. No fluxograma abaixo apresentado são indicadas as etapas do processamento desta informação para, de seguida, serem explicados os métodos e algoritmos aplicados.

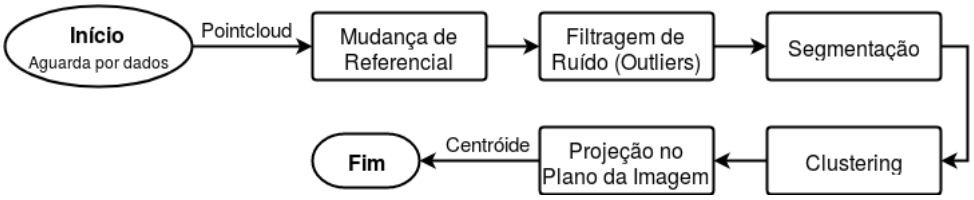

Figura 3.6: Fluxograma das etapas de processamento da *pointcloud*

Fisicamente, é impossível a coexistência de dois sensores na mesma localização espacial. Isto leva a que os dados obtidos se encontrem em referenciais diferentes, com diferentes perspetivas. Para que seja possível a junção da informação de ambos, é necessária que esta seja convertida para um único referencial, pelo que optou-se pela transformação dos dados do LiDAR para o referencial da câmara. Para este processo utilizou-se os parâmetros extrínsecos da câmara em relação ao LiDAR, que consistem na posição e na orientação relativa entre os seus referênciais. Assim, os novos pontos são dados pela expressão matemática 3.1,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T \tag{3.1}$$

em que R é a matriz de rotação, (X,Y,Z) são as coordenadas do ponto no referencial do LiDAR e T a matriz de translação. Esta informação está disponível, graças à prévia calibração do sistema

aquando da sua montagem.

$$R = \begin{bmatrix} 0.02322 & -0.99965 & -0.01245 \\ 0.02312 & 0.01299 & -0.99965 \\ 0.99946 & 0.02293 & 0.02341 \end{bmatrix} \qquad T = \begin{bmatrix} 0.10335 \\ -0.23752 \\ 0.05256 \end{bmatrix} \tag{3.2}$$

Após a recolha dos dados e passagem para o novo referencial, as próximas etapas prendem-se com a procura de aglomerados de pontos que possam representar objetos. Para alcançar tal objetivo, inicialmente é feita uma filtragem da *pointcloud* para remoção do ruído. Depois, uma segmentação da representação em planos, para que pontos que representem o solo sejam removidos. Por fim, os pontos que sobram são agrupados em *clusters* cuja métrica de aglomeração é baseada na distância euclidiana. Desta forma, pertencem ao mesmo aglomerado todos os pontos que se encontrem a menos de uma determinada distância da sua vizinhança.

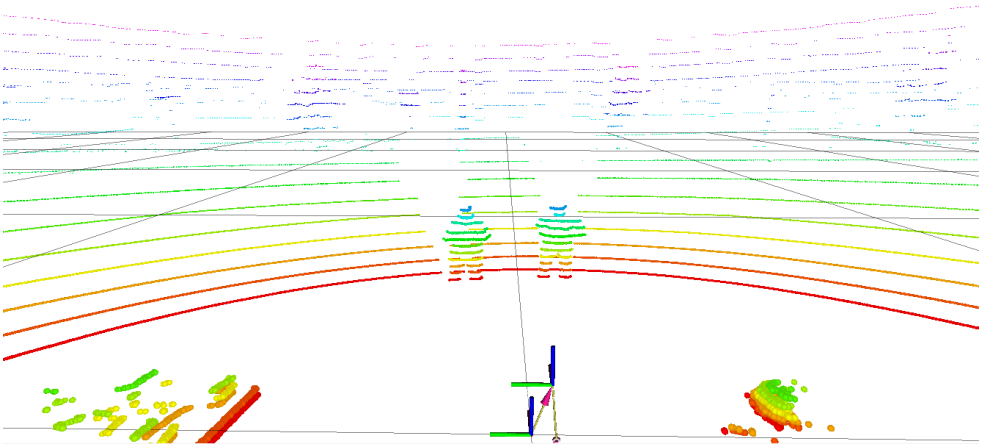

Figura 3.7: *Pointcloud* recebida

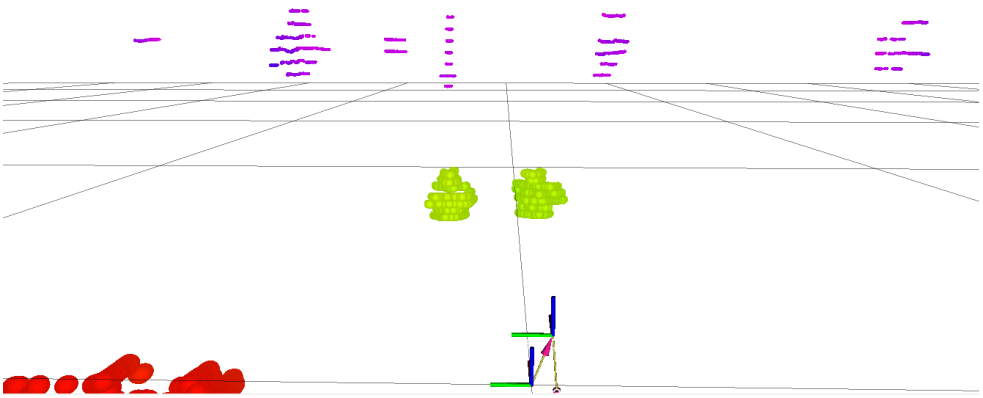

Figura 3.8: *Pointcloud* após o processamento

Uma vez que cada um destes *clusters* é, potencialmente, um objeto a ter em consideração, é calculado o seu centróide para posterior utilização num processo de associação espacial e temporal com os objetos encontrados na imagem. Todas estas operações são realizadas recorrendo a funções disponíveis na biblioteca PCL.

O ponto final no processamento da *pointcloud* dá-se com a projeção dos pontos no plano da imagem. Uma vez que de um lado os dados são tridimensionais e do outro bidimensionais, com representação em coordenadas de pixeis, é necessária a conversão destes para a mesma dimensão para que seja possível, depois, a associação destes dados. É utilizado o modelo de câmara *pinhole*, que descreve a relação matemática entre as coordenadas de um ponto tridimensional e a sua projeção no plano de uma imagem.

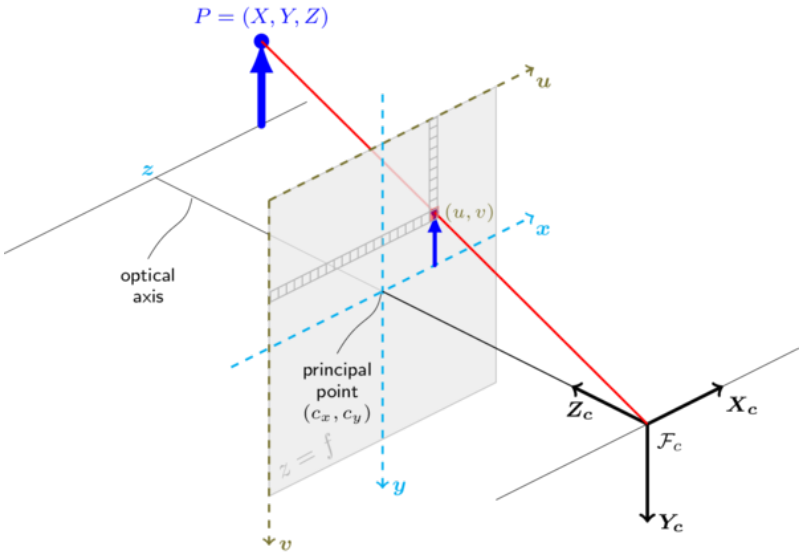

Figura 3.9: Projeção de um ponto tridimensional no plano de uma imagem

Para a obtenção das coordenadas 2D, são utilizadas as expressões matemáticas 3.1, 3.3 e 3.4. A expressão 3.1 volta a surgir, apesar de os pontos encontrarem-se já no referencial da câmara,uma vez que estes devem ser transformados para o referencial da imagem, que neste caso é diferente do sensor.

$$\begin{cases} x' = x/z \\ y' = y/z \end{cases} \tag{3.3}$$

$$\begin{cases} u = f_x * x' + c_x \\ v = f_y * y' + c_y \end{cases} \tag{3.4}$$

em que (x,y,z) são as coordenadas tridimensionais do ponto no referencial do sensor (câmara) e (u,v) as coordenadas em pixel do ponto bidimensional projetado no plano. Para além dessas, surgem os parâmetros:

- (f_x,f_y) - distância focal, expressa em pixeis;

- (c_x,c_y) - ponto central da imagem que define o eixo ótico da câmara.

Estes parâmetros são retirados da matriz dos parâmetros intrínsecos da câmara, K, cujos valores são calibrados e dependentes exclusivamente das características físicas do conjunto câmara e lente.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1610.89 & 0 & 649.48 \\ 0 & 1615.49 & 464.75 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Após a conversão os pontos tridimensionais cujas coordenadas projetadas se encontrem dentro do plano da imagem são armazenados numa matriz tripla. Neste caso, como a resolução das imagens captadas é de 1292x964, os pontos com coordenadas (u,v) entre (0,0) e (1291,963).

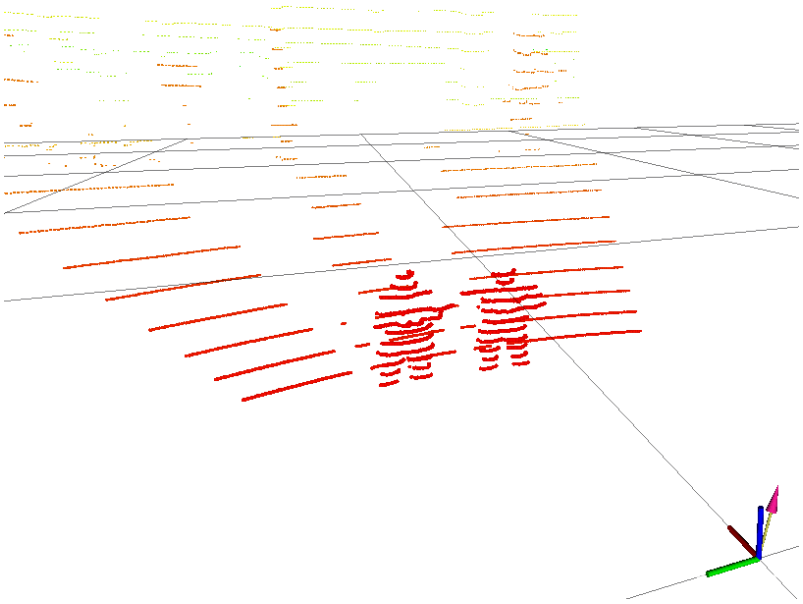

Figura 3.10: Pontos enquadrados com o plano da imagem

Esta matriz está dividida em três camadas, uma para cada componente dos pontos (x,y,z). Servirá de conexão entre os objetos detetados na imagem (a duas dimensões) e a representação do LiDAR, uma vez que será utilizada para fazer a correspondência entre o centro desses objetos e um ponto tridimensional.

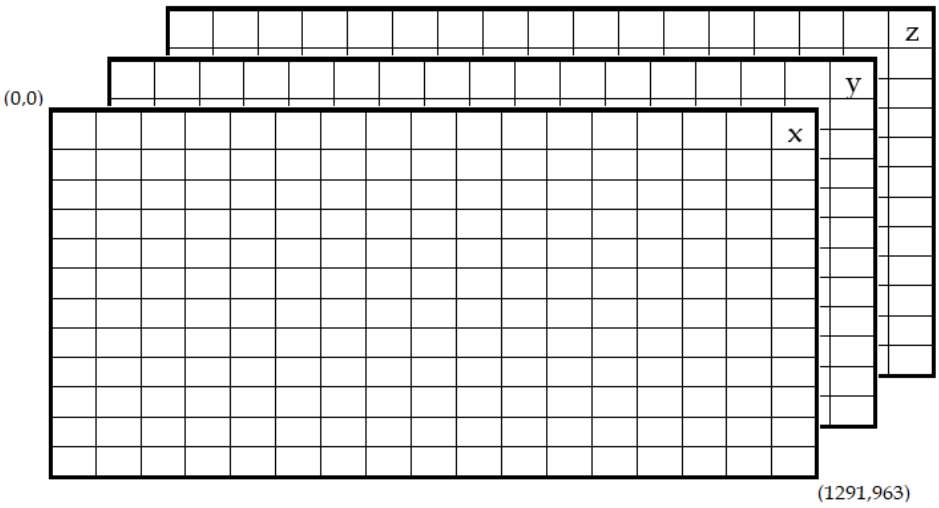

Figura 3.11: Representação da estrutura matricial armazenadora dos pontos 3D projetados

3.3.3 Associação

Com o processo de deteção de objetos concluído e com os *clusters* extraídos da *pointcloud*, prossegue-se agora com dois processos de associação, a associação espacial e a temporal. O objetivo da primeira é obter a correspondência entre os objetos detetados e os *clusters*. Já o processo de associação temporal pretende associar aos objetos encontrados (os candidatos) às entidades existentes no instante anterior, para que seja possível o cálculo das características dinâmicas (velocidade e direção).

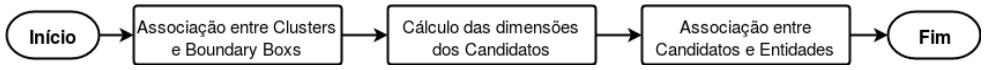

Figura 3.12: Fluxograma das etapas de associação

Devido a incertezas nos valores de calibração da câmara e nas matrizes de rotação e translação entre referenciais, ocorrem erros na projeção dos pontos tridimensionais para o plano da imagem. Como consequência, por vezes, o ponto central da *boundary box* do objeto não corresponde a um ponto pertencente a um *cluster*. De forma a garantir a associação espacial, é então calculada a distância euclidiana entre os centróides dos *clusters* existentes e o ponto. É atribuída a correspondência à distância menor, desde que esta seja inferior a um *threshold* de cinquenta centímetros. Assim são acauteladas situações em que não é extraído qualquer aglomerado de pontos na periferia prevista do ponto. Realizando esta operação para todos os objetos detetados na imagem, obtém-se uma lista de candidatos ao título de entidade.

Antes da associação entre as entidades existentes e os candidatos da presente iteração, são calculadas as dimensões tridimensionais aproximadas dos candidatos. Este valor é facilmente adquirido sabendo a distância do sensor ao objeto e as dimensões da sua *boundary box*, recorrendo

ao método da triangulação para determinar a localização tridimensional das extremidades dos objetos, como referido no capítulo 2. Os candidatos, identificados com uma classe (tipo de objeto), um valor médio de cor, com a sua localização tridimensional relativa ao sistema e com as suas dimensões, estão assim prontos para a etapa da associação temporal.

A associação temporal, como referida anteriormente, permitirá a correspondência entre entidades detetadas nos instantes anteriores e os candidatos observados na iteração atual. Esta etapa dará ferramentas à aplicação para determinar o deslocamento das entidades ao longo de instantes consecutivos de tempo e atualizar a sua perceção do ambiente, atualizando a sua lista de entidades no caso de alguma desaparecer ou de uma nova entidade surgir.

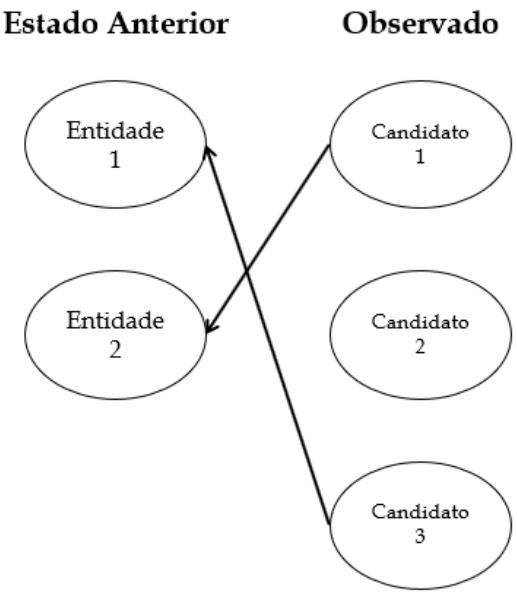

Figura 3.13: Exemplo associação entre entidades e candidatos

Para que se concretize esta associação é utilizada uma função de semelhança, que procede ao calculo do somatório das diferenças entre características das entidades e candidatos para fazer a correspondência, como apresentado na expressão 3.6. Aos somatórios com os valores mais baixos são atribuídas correspondências.

$$W = \sum d_i \tag{3.6}$$

À diferença entre os valores da entidade e do candidato atribuiu-se a variável d_i. As características a ter em conta são a diferença na dimensão dos objetos e a diferença entre a cor média, assim como a distância euclidiana entre entidade e candidato. Para o cálculo desta última, deve-se ter em especial atenção a possibilidade da mudança de posição entre a captura dos pontos nas duas iterações, devido ao movimento do veículo no ambiente, já que o sistema em desenvolvimento tem em mente robôs dotados de mobilidade. Utilizando os dados GPS é calculado o deslocamento do

sistema, para que seja aplicado na localização da entidade e, assim, ser possível a determinação da distância euclidiana no mesmo referencial.

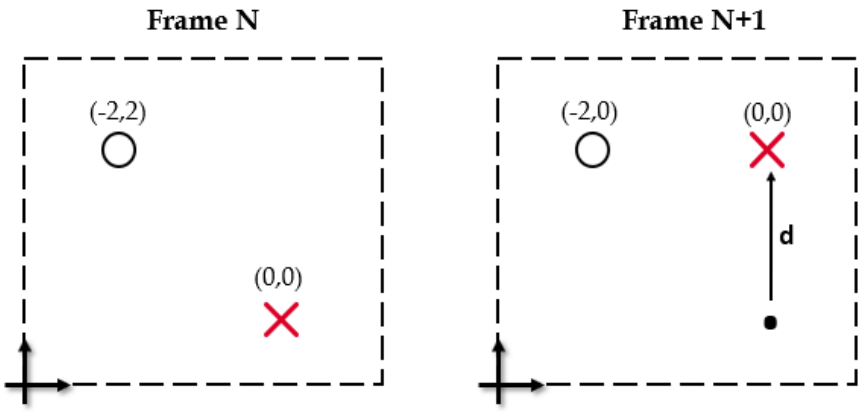

Figura 3.14: Posição relativa de um objeto ao sistema

Determinadas todas as associações, são atualizadas as características das entidades com os valores mais recentes, retirados dos candidatos. Para além disso, é atualizada a lista de objetos, aumentando ou diminuindo conforme a existência de entidades ou candidatos por associar. O cruzamento de duas entidades perante o sistema de perceção, obstáculos ou até a passagem de outras entidades mais próximas dos sensores, podem provocar momentaneamente a oclusão parcial ou completa de alguns objetos. Para evitar a perda momentânea dessas entidades, uma vez que podem não ser identificadas nessas ocasiões, é implementado um sistema de *fadeout*. Desta forma, se não existir correspondência para uma entidade, será procurada a existência de um *cluster* suficientemente próximo da localização do objeto,entre os *clusters* sem associação a candidatos. Se houver, não é eliminada imediatamente da lista mas contudo, como o grau de confiança na identificação é menor, é diminuída a sua temperatura. Assim, é apenas removida quando o grau de confiança atinge o valor zero.

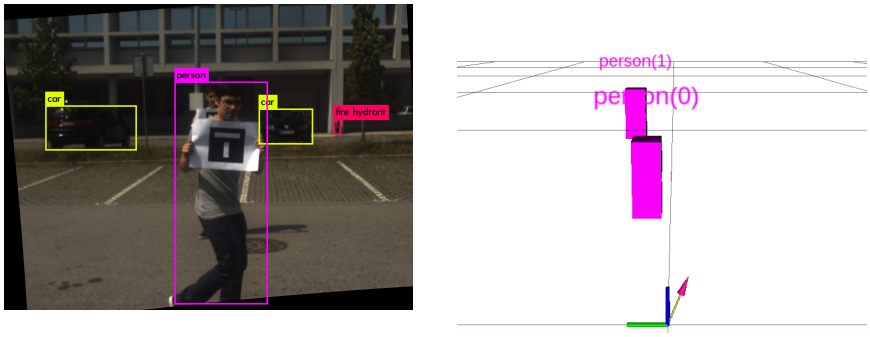

Figura 3.15: Situação de oclusão, com contínuo acompanhamento do sistema

3.3.4 Características Dinâmicas dos Objetos

Mais do que a perceção, a capacidade de previsão das movimentações do ambiente envolvente permitem que o sistema possa acautelar ocorrências que se poderiam tornar perigosas para a sua operação, como o caso de colisões. Assim sendo, é importante que o sistema conheça as características dinâmicas das entidades que o rodeiam, a velocidade e a direção, para que possa prever o seu movimento e assim aferir se estão ou não em rota de colisão com o veículo e se, por isso, é necessário tomar medidas preventivas.

Uma vez que na etapa anterior é processado o deslocamento em vetor e módulo, resta agora obter o intervalo temporal entre iterações para descobrir a velocidade média do objeto, já que esta é dada pela expressão 3.7, sendo a razão entre o deslocamento e o intervalo de tempo ocorrido entre as medições da localização. Quanto à direção, é obtida da normalização do vetor deslocamento.

$$v = \frac{\Delta s}{\Delta t} \tag{3.7}$$

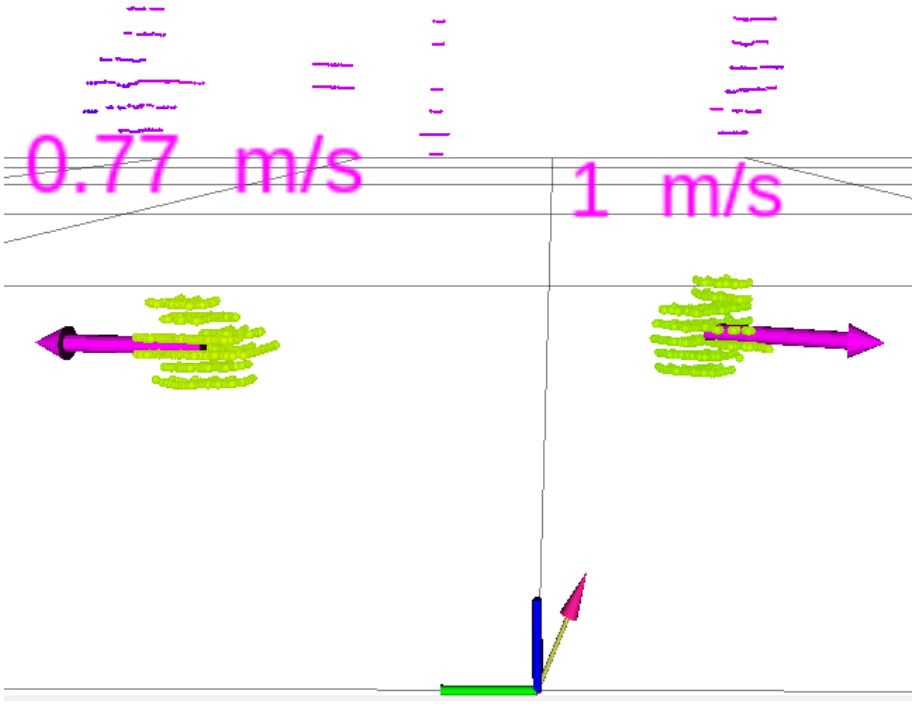

Figura 3.16: Representação da direção e velocidade

Com esta etapa, a entidade fica completamente caracterizada, espacial e temporalmente. Sempre que novos dados chegam, todas as etapas descritas anteriormente são repetidas, originando um sistema capaz de acompanhar o desenvolvimento do ambiente envolvente e das movimentações nele.

3.4 Resultados

Para a validação do sistema em desenvolvimento foram realizados testes utilizando dados recolhidos, para o efeito, pela equipa do CRAS. Optou-se pela gravação de dados em cenários terrestres, em que objetos como carros e pessoas são captados pelos sensores, tanto em movimento como imobilizados. De seguida, serão apresentados os dados resultantes de cada uma das etapas do processo, em diferentes cenários, e serão feitos alguns comentários e considerações sobre os resultados esperados e os conseguidos.

3.4.1 Análise da deteção e classificação de objetos

Como referido em 3.3, optou-se pela utilização do projeto YOLO para a deteção e classificação dos objetos. Segundo os seus autores, recorrendo à sua rede neuronal mais pequena é possível processar até 120FPS e, com a rede neuronal mais completa e assertiva, 45FPS. Assim, no pior dos casos e segundo as suas afirmações, seria possível a classificação dos objetos numa imagem com uma latência inferior a 25 milissegundos, o que seria perfeito para aplicações em tempo real. Por isso, o primeiro passo tomado passou pela averiguação do tempo de processamento com o equipamento disponível no laboratório e a qualidade dos resultados, para que existisse confiança na aplicação deste projeto numa etapa fulcral do sistema.

	CPU				GPU			
	YOLO	σ	YOLO Tiny	σ	YOLO	σ	YOLO Tiny	σ
Tempo de Processamento (s)	11.703	0.2847	0.9347	0.0645	0.1662	0.0063	0.1034	0.0010
FPS	0.09		1.07		6.02		9.67	

Tabela 3.2: Tempos de processamento médio dos vários métodos aplicáveis ao processamento do YOLO

Como se pode observar pela tabela, os resultados foram bem inferiores aos esperados. Contudo, é necessário ter em atenção que o equipamento existente é inferior ao utilizado nos testes dos autores, com uma unidade de processamento gráfico com menor poder, sendo possivelmente esta uma das causas na diferença dos valores. Apesar da diferença, com cerca de nove imagens processadas por segundo, seria possível a criação de um sistema que funcionasse em tempo real, mas com a contra-partida de os seus resultados serem mais intervalados e com isso surgirem problemas na deteção e caracterização de objetos a maior velocidade. Para a aplicação nesta dissertação optou-se então pelo pré-processamento das imagens e utilização dos seus resultados numa simulação de tempo real.

Quanto à qualidade das classificações, nos testes realizados com diferentes cenários e condições de luz, os resultados tomaram valores muito positivos. Em todas as frames observadas, os objetos detetados foram corretamente classificados, mesmo a distâncias consideráveis. Contudo,

em ambientes muito preenchidos, com muitos objetos de pequenas dimensões, agrupados, nem todos os objetos foram detetados. Alguns dos resultados obtidos são, de seguida, apresentados.

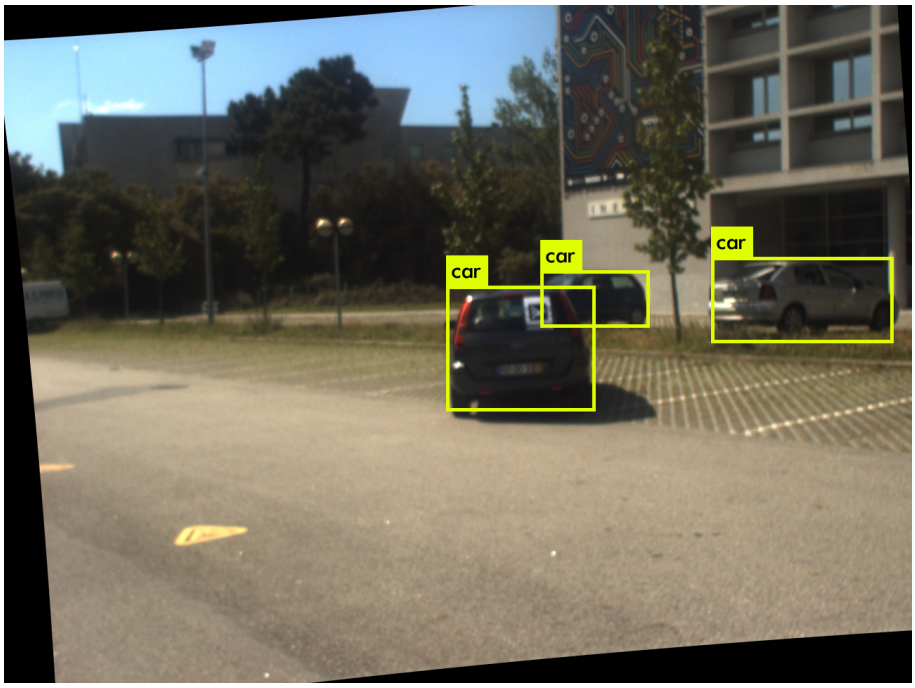

Figura 3.17: Classificação de objetos em ambiente iluminado, com oclusão parcial

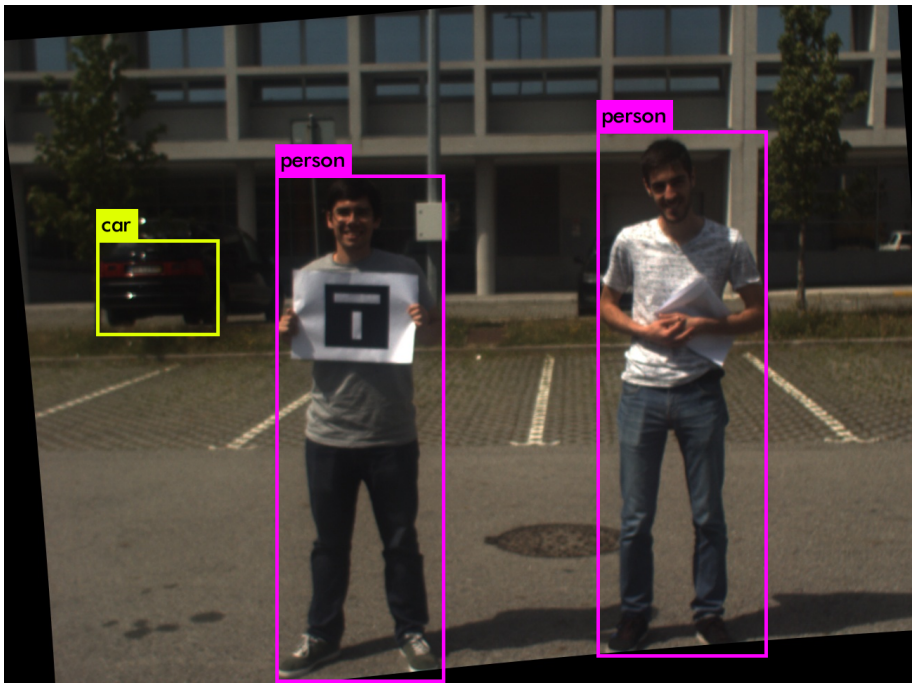

Figura 3.18: Classificação de objetos em ambiente com iluminação moderada

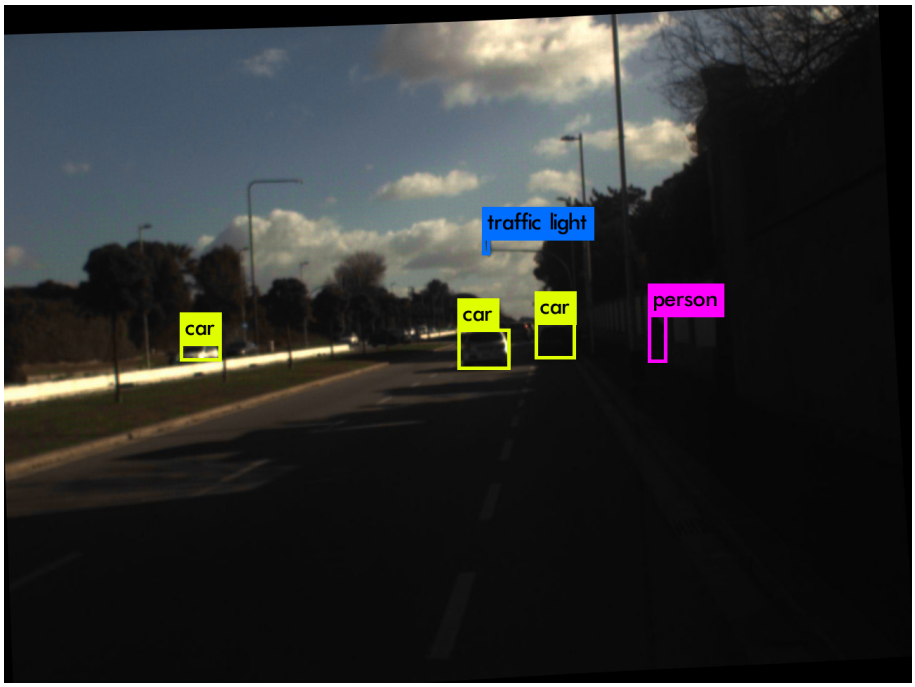

Figura 3.19: Classificação de objetos em ambiente pouco iluminado e a grandes distâncias

Mesmo em ambientes mais povoados são conseguidos resultados positivos. Na figura 3.20 é possível observar um ambiente urbano, repleto de viaturas, na qual foi aplicada a deteção com sucesso.

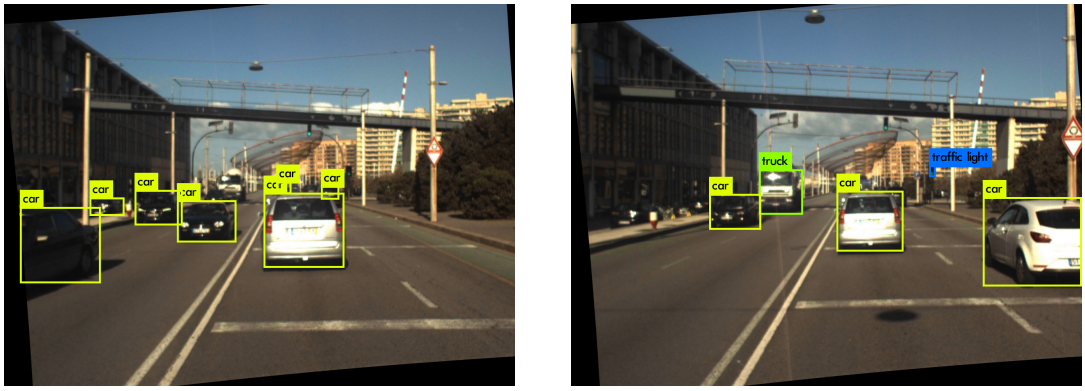

Figura 3.20: Classificação em ambientes com tráfego moderado

Concluindo a apresentação dos resultados desta etapa, convém sublinhar que a entrave principal à aplicação deste projeto a um sistema autónomo prende-se à necessidade de uma unidade de processamento muito poderosa, que além do espaço que ocuparia e do custo que impingiria ao sistema, exige um consumo de energia que pode ser insuportável em sistemas que se querem operacionais durante longos períodos de tempo.

3.4.2 Cenário 1: pessoas num parque de estacionamento

O primeiro cenário escolhido para testar o sistema foi um ambiente controlado, no qual duas pessoas foram entrando e saindo do ângulo de visão da câmara. Como fundo, um pequeno parque de estacionamento, no qual existem carros parados. O uso deste cenário pretende testar a capacidade de deteção e conjugação dos dados da câmara e do LiDAR, querendo verificar-se se são observados corretamente, ao longo do tempo, os objetos das diferentes classes.

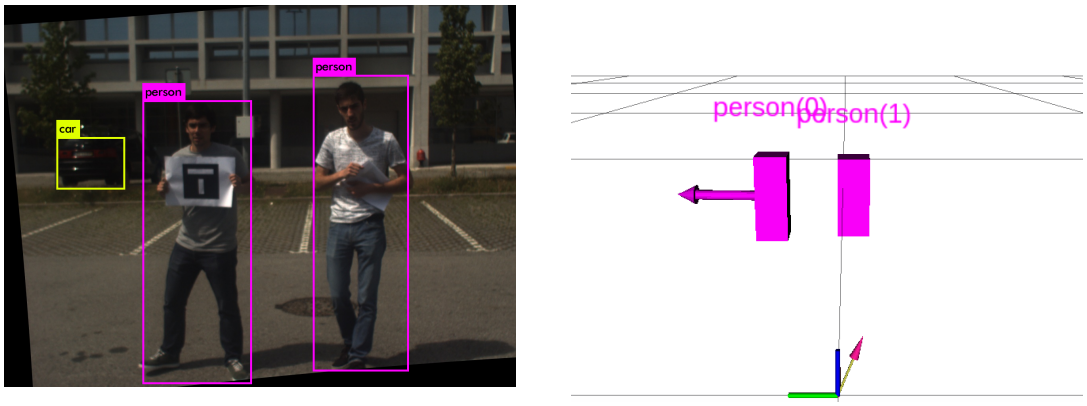

Figura 3.21: Cenário 1 - Momento 1

Nos momentos iniciais surgem diante da câmara duas pessoas, sendo possível observar no fundo, à esquerda, um carro. Estas três entidades são identificadas pelo YOLO contudo, como é possível averiguar na Figura 3.21, o carro não é representado pelo sistema. Após averiguação, concluiu-se que este erro se deve à qualidade dos dados obtidos pelo LiDAR. Observando-os, é possível notar que a densidade dos pontos captados a maior distâncias é baixa, o que dificulta o *clustering* destes. Assim, como se pode ver na Figura 3.8, o sistema não foi capaz de agrupar os pontos do carro, sendo que o único *cluster* na área corresponde à árvore à esquerda deste.

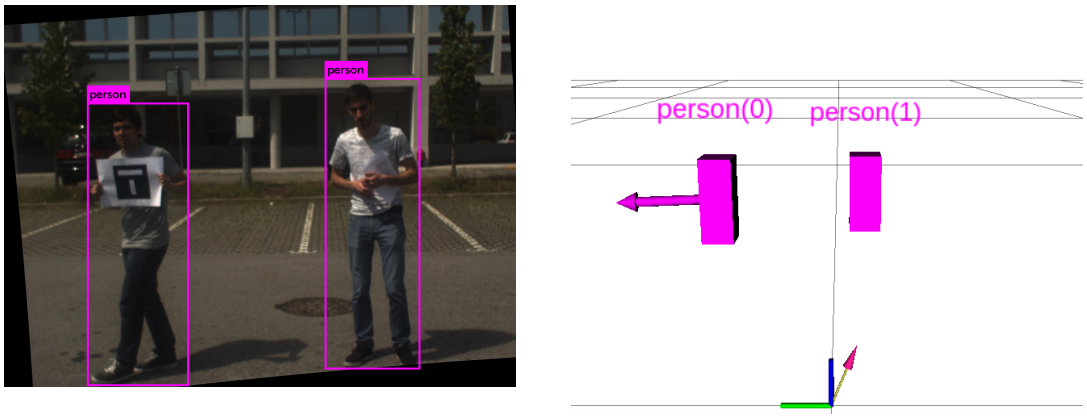

Figura 3.22: Cenário 1 - Momento 2

Os momentos seguintes deste cenário são aproveitados para comprovar as etapas de caracterização e associação temporal do sistema. Como referido noutros momentos da dissertação, cada

um destes objetos tem associado, para além da classificação, dimensões, velocidade e direção. Ao longo da presente sequência de imagens é possível observar a forma como o sistema atribui sistematicamente, à pessoa da esquerda (no Momento 1 e 2) o número de identificação zero e à pessoa da direita o número um. Além disso, é possível observar que o movimento de cada uma destas entidades é acompanhado, sendo representado na forma de uma seta, que aponta na direção do movimento.

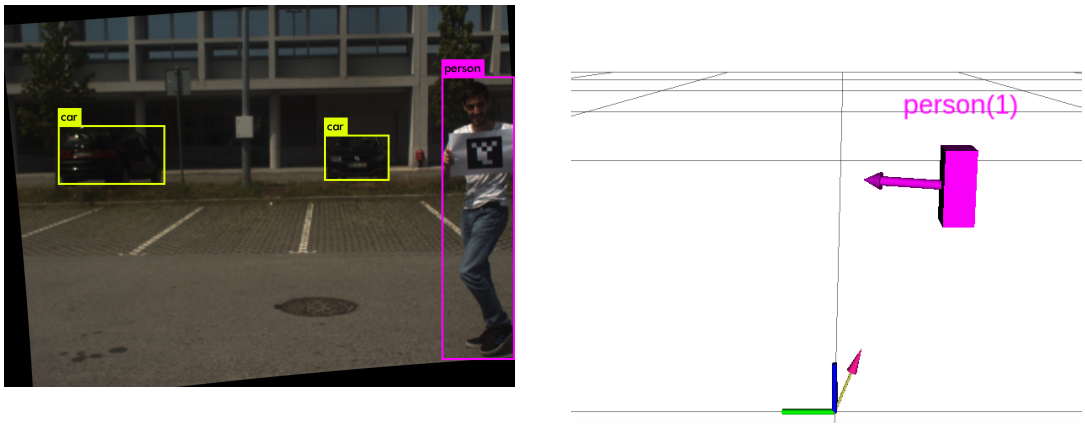

Figura 3.23: Cenário 1 - Momento 3

Quanto às dimensões e velocidade calculadas, os valores obtidos são bastante satisfatórios. Tendo em conta a altura de cada uma das pessoas detetadas, as dimensões apresentadas oscilaram em valores com erro entre os dois e os dez centímetros. Já as velocidades calculadas estão, segundo [34], entre os valores registados como normais para o ato de caminhar, com velocidades entre os 0.7 e os 1.5 m/s.

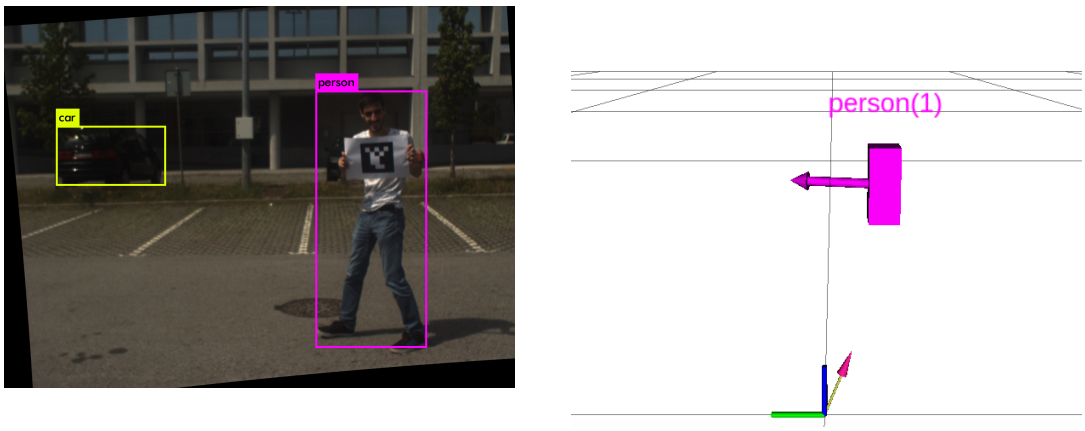

Figura 3.24: Cenário 1 - Momento 4

Por fim, nos momentos cinco a oito, a aplicação é presenteado com a oclusão temporária de uma das entidades, de forma a testar o sistema de *fadeout* implementado. Assim, e como observado, ambas as entidades são continuamente representadas, como pretendido. De notar que, no

momento cinco (Figura 3.25) ocorrem dois falsos positivos, em que um dos *clusters* encontrados é atribuído ao carro na esquerda da imagem e outro, à direita, é atribuído a uma boca de incêndio detetada pelo YOLO. Apesar de estes coincidirem com a direção destes objetos, esta identificação não passa de uma feliz coincidência, uma vez que os pontos em causa correspondem às árvores nos extremos da imagem.

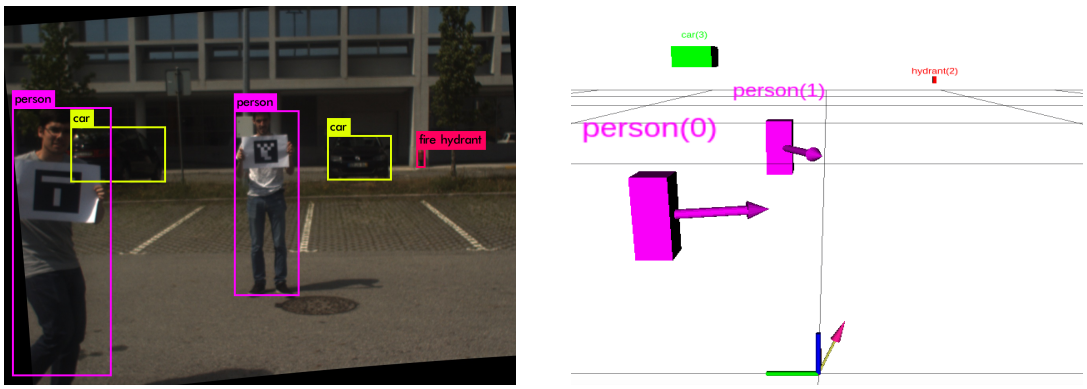

Figura 3.25: Cenário 1 - Momento 5

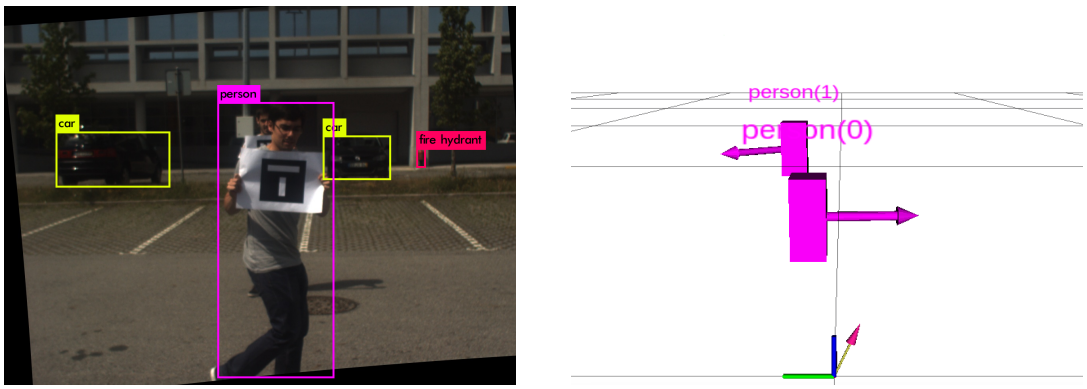

Figura 3.26: Cenário 1 - Momento 6

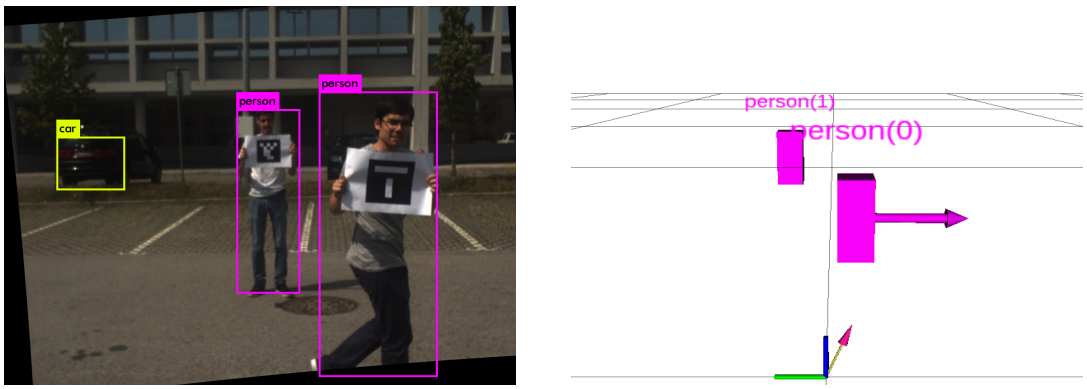

Figura 3.27: Cenário 1 - Momento 7

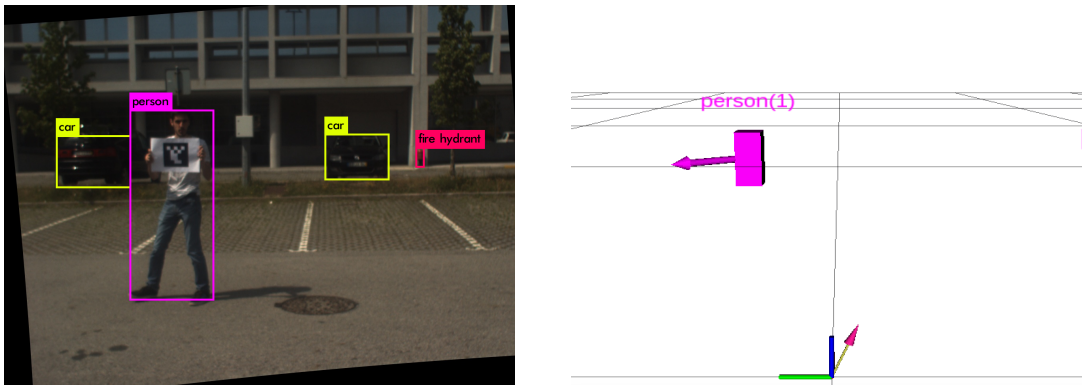

Figura 3.28: Cenário 1 - Momento 8

A aplicação do sistema neste cenário permitiu encontrar debilidades e restrições à sua utilização, sendo um fator negativo a incapacidade deste em acompanhar objetos que se encontrem a maiores distâncias. Para colmatar este fator, seria necessário aumentar a densidade de pontos captados, seja através da junção de *pointclouds* de múltiplos LiDARs ou recorrendo a um sensor com maior alcance e maior número de pontos.

3.4.3 Cenário 2: pessoas no jardim

No segundo cenário apresentado, um total de cinco pessoas coloca-se perante os sensores, pretendendo-se averiguar como o sistema se comporta com múltiplas entidades em movimento, com velocidades distintas. Nos momentos iniciais, surge uma pessoa que se desloca perpendicularmente ao plano da imagem. Uma vez que no cenário anterior existiram dificuldades em detetar objetos que estivessem a maiores distâncias, esta entidade será utilizada para desafiar o alcance do sistema, afastando-se continuamente deste.

Figura 3.29: Cenário 2 - Momento 1

Figura 3.30: Cenário 2 - Momento 2

De seguida, nos momentos três e quatro, ocorre o cruzamento de duas pessoas perante o sensor, ocultando temporariamente a primeiro pessoa em cena. Apesar desta oclusão, as três entidades são corretamente identificadas.

Figura 3.31: Cenário 2 - Momento 3

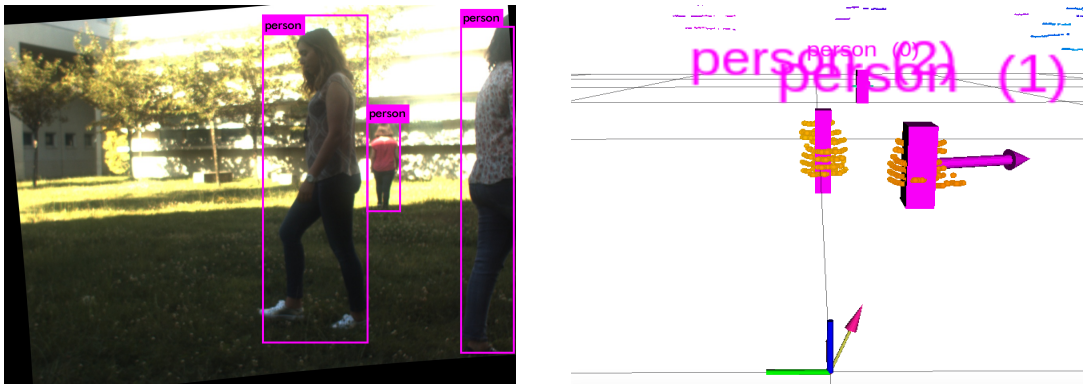

Figura 3.32: Cenário 2 - Momento 4

Nos momentos cinco e seis é testada a reação do sistema a uma entidade que se desloque a maior velocidade, neste caso uma pessoa a correr. A primeira identificação deste dá-se na Figura 3.33, na qual já se encontra no centro da imagem. Se o sistema estivesse em movimento, haveria

o risco de ocorrer a colisão com esta entidade, uma vez que a identificação dá-se já muito tarde. Para tornar o sistema mais seguro seria necessário, por isso, diminuir o seu tempo de resposta, quer através da otimização dos algoritmos aplicados quer da utilização de *hardware* mais poderoso.

Figura 3.33: Cenário 2 - Momento 5

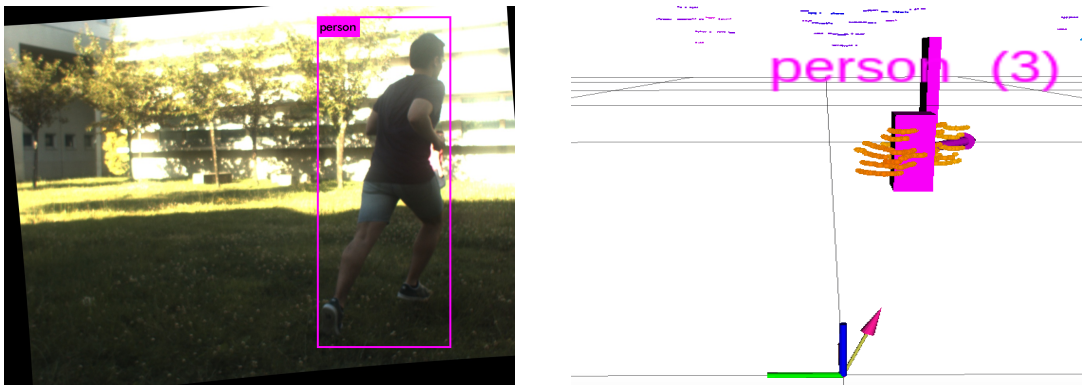

Figura 3.34: Cenário 2 - Momento 6

Por fim, nos momentos sete e oito, uma pessoa cruza o plano da imagem, percorrendo o cenário diagonalmente, sendo identificado e representado corretamente o seu deslocamento.

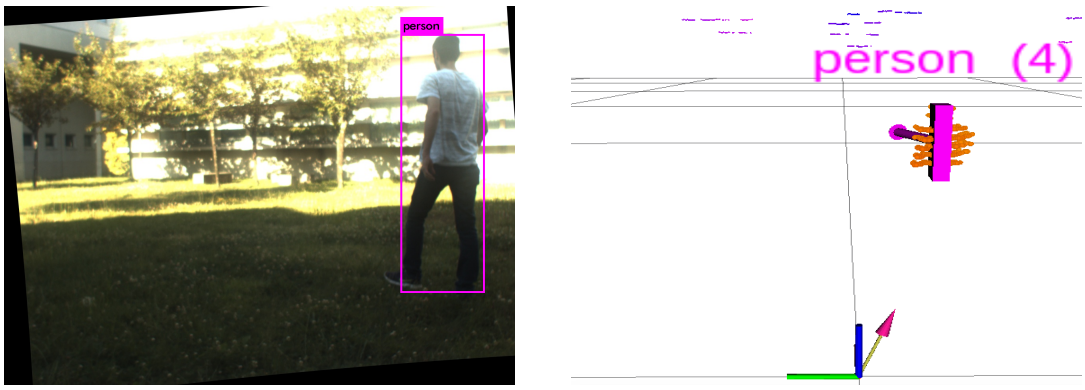

Figura 3.35: Cenário 2 - Momento 7

Figura 3.36: Cenário 2 - Momento 8

Da aplicação do sistema neste cenário, é necessário referir algumas particularidades notadas. Quanto à primeira pessoa em deslocamento e o alcance do sistema, esta foi detetada até a uma distância de cerca de dez metros. Se se estivesse perante um carro em trajetória de colisão, a uma velocidade de 50Km/h, o sistema teria menos de um segundo para reagir e mudar a sua direção. Assim, pode-se concluir que a distância alcançada não é suficiente para uma navegação completamente segura do veículo, pelo menos para aplicação em meios com veículos que possam alcançar velocidades parecidas ou superiores. Notou-se também que com o aumentar da distância tornou-se menos precisa a medição das dimensões, sendo que inicialmente o erro oscilou entre dois a cinco centímetros e, a dez metros de distância, oscilou entre cinco a trinta centímetros. Por fim, e como já referido, a deteção de entidades de maior velocidade não é conseguida com velocidade suficiente, mesmo quando estas se encontram perto do sistema. É necessário otimizar os algoritmos criados para que o processamento desta informação seja mais eficiente.

Capítulo 4

Conclusões e Trabalho Futuro

4.1 Conclusão

O número de robôs que têm surgido para auxiliar ou substituir os humanos, nas mais diversas aplicações e tarefas, tem aumentado a passos largos. Inerente ao seu desenvolvimento, surge a necessidade de dotá-los com capacidades de percepção avançadas, para que a interação do sistema com o ambiente corra sem percalços. Em aplicações de vigilância acresce, para além da identificação de obstáculos, o interesse na caracterização tridimensional detalhada dos objetos encontrados. É interessante a capacidade de distinguir os objetos móveis dos estáticos, identificar o tipo de objetos perante os quais estão, a sua localização, dimensões, velocidade e direção. Desta forma, ao longo da presente dissertação, foi desenvolvido um sistema que pretende retirar do ambiente toda esta informação e colocá-la à disposição do utilizador.

Uma das particularidades fundamentais na sua criação prendeu-se com a capacidade que este deveria ter de, em tempo real, fornecer informações precisas destas características. Assim, de forma a colmatar erros e imprecisões, optou-se pela incorporação da informação de múltiplos sensores para a realização desta tarefa.

Inicialmente procurou-se projetos realizados no âmbito da identificação e classificação de objetos para aplicação no sistema, uma vez que, por si só, esta área é estudada por uma larga comunidade científica e o foco desta dissertação são as características tridimensionais. No processo de decisão pesaram, para além da capacidade para funcionar em tempo real, a *performance* dos algoritmos e as restrições das suas classificações, já que o sistema torna-se mais interessante se poder acompanhar um número maior de objetos que vão para além de carros e peões. A identificação de navios e animais marítimos, por exemplo, pode permitir o transporte desta aplicação para ASVs, para a realização de tarefas de patrulhamento da orla costeira. A escolha recaiu no YOLO, projeto *open source* capaz de detetar e classificar milhares de objetos em imagens, com latência de milissegundos, mesmo perante fracas condições de iluminação e a grandes distâncias.

Após a deteção, adicionou-se ao sistema uma forma mais exata do que a obtida por imagens para a localização dos objetos, utilizando a informação de um LiDAR. A sua representação

tridimensional do ambiente, associada à deteção feita pelo YOLO, permitiu a localização tridimensional dos objetos relativamente ao sistema com apenas alguns centímetros de erro. Assim, acompanhando o posicionamento destes ao longo do tempo, foi possível a determinação da velocidade e direção com exatidão. Contudo verificaram-se, durante os testes aplicados, erros na associação da informação destes sensores. O alcance do LiDAR, devido à dispersão dos pontos com o aumento da distância ao referencial, tornou-se pequeno para a capacidade de deteção a longas distâncias do YOLO, o que tornou impossível, em algumas ocasiões, a caracterização de alguns objetos detetados e, noutras, invalidou os valores apresentados.

O método de identificação e caracterização desenvolvido demonstrou potencial nos testes efetuados, obtendo estimações das características pretendidas com exatidão. Porém, para que seja possível a sua transição para um sistema em ambiente e tempo real, é ainda necessário, como os testes poderam comprovar, a otimização dos processos para que o número de objetos que fica por caracterizar não ponha em risco a operação do sistema.

4.2 Trabalho futuro

O trabalho desenvolvido nesta dissertação cumpriu com os objetivos inicialmente propostos. No entanto, como referido em 4.1, necessita de otimizações para que possa ser aplicado em ambiente real. Dessa forma, propõe-se como sugestão de trabalho futuro as seguintes tarefas:

- modificação do *hardware* utilizado para colmatar a baixa densidade de pontos existentes a longa distância, obtidos com o LiDAR
- otimização dos algoritmos implementados para aumentar a eficiência computacional
- inclusão dos pontos da *pointcloud* para melhorar a determinação das dimensões dos objetos
- realização de testes em ambiente marítimo

Referências

[1] Gang Wang, Yongdong Zhang, e Jintao Li. High-level background prior based salient object detection. *Journal of Visual Communication and Image Representation*, 48:432–441, 2017. URL: http://dx.doi.org/10.1016/j.jvcir.2017.02.004, doi:10.1016/j.jvcir.2017.02.004.

[2] How to achieve corner detection in C#. URL: http://www.camera-sdk.com/p{_}256-how-to-accomplish-corner-detection-in-c-onvif.html.

[3] Alireza Asvadi, Cristiano Premebida, Paulo Peixoto, e Urbano Nunes. 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems*, 83:299–311, 2016. URL: http://dx.doi.org/10.1016/j.robot.2016.06.007, doi:10.1016/j.robot.2016.06.007.

[4] MathWorks. Exchange Data with ROS Publishers and Subscribers. URL: https://www.mathworks.com/help/robotics/examples/exchange-data-with-ros-publishers-and-subscribers.html.

[5] Jonathan Hui. mAP (mean Average Precision) for Object Detection. URL: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

[6] Joseph Redmon e Ali Farhadi. Yolov2. URL: https://pjreddie.com/darknet/yolov2/.

[7] Manisha Kaushal, Baljit S Khehra, e Akashdeep Sharma. Soft Computing based object detection and tracking approaches: State-of-the-Art survey. *Applied Soft Computing Journal*, 70:423–464, 2018. URL: https://doi.org/10.1016/j.asoc.2018.05.023, doi:10.1016/j.asoc.2018.05.023.

[8] Mehran Yazdi e Thierry Bouwmans. New trends on moving object detection in video images captured by a moving camera: A survey. *Computer Science Review*, 28:157–177, 2018. URL: https://doi.org/10.1016/j.cosrev.2018.03.001, doi:10.1016/j.cosrev.2018.03.001.

[9] Asma Azim. 3D Perception of Outdoor and Dynamic Environment using Laser Scanner. páginas 150–176, 2013.

[10] Peixia Li, Dong Wang, Lijun Wang, e Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018. URL: https://doi.org/10.1016/j.patcog.2017.11.007, doi:10.1016/j.patcog.2017.11.007.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, e Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. 2015. URL: http://arxiv.org/abs/1506.02640, arXiv:1506.02640, doi:10.1109/CVPR.2016.91.

[12] Heng Fan e Haibin Ling. SANet: Structure-Aware Network for Visual Tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:2217–2224, 2017. arXiv:1611.06878, doi:10.1109/CVPRW.2017.275.

[13] Yaran Chen, Dongbin Zhao, Le Lv, e Qichao Zhang. Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432:559–571, 2018. doi:10.1016/j.ins.2017.08.035.

[14] John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, Olivier Koch, Yoshiaki Kuwata, David Moore, Edwin Olson, Steve Peters, Justin Teo, Robert Truax, Matthew Walter, David Barrett, Alexander Epstein, Keoni Maheloni, Katy Moyer, Troy Jones, Ryan Buckley, Matthew Antone, Robert Galejs, Siddhartha Krishnamurthy, e Jonathan Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, oct 2008. URL: http://doi.wiley.com/10.1002/rob.20262, arXiv:10.1.1.91.5767, doi:10.1002/rob.20262.

[15] Vasundhara G. Posugade e Rohita P. Patil. FPGA based design and implementation of disparity estimation for stereo vision system. *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, páginas 1–5, 2016. URL: http://ieeexplore.ieee.org/document/7860050/, doi:10.1109/ICCUBEA.2016.7860050.

[16] Serdar Solak e Emine Dogru Bolat. Distance Estimation using Stereo Vision for Indoor Mobile Robot Applications. *IEEE - 9th International Conference on Electrical and Electronics Engineering (ELECO) 2015*, páginas 685–688, 2015. doi:10.1109/ELECO.2015.7394442.

[17] J. C. Rodríguez-Quiñonez, O. Sergiyenko, W. Flores-Fuentes, M. Rivas-lopez, D. Hernandez-Balbuena, R. Rascón, e P. Mercorelli. Improve a 3D distance measurement accuracy in stereo vision systems using optimization methods' approach. *Opto-electronics Review*, 25(1):24–32, 2017. doi:10.1016/j.opelre.2017.03.001.

[18] Ondrej Kainz, Frantisek Jakab, Matus W. Horecny, e David Cymbalak. Estimating the object size from static 2D image. *2015 International Conference and Workshop on Computing and Communication, IEMCON 2015*, 2015. doi:10.1109/IEMCON.2015.7344423.

[19] Serdar SOLAK e Emine Doğru BOLAT. A new hybrid stereovision-based distance-estimation approach for mobile robot platforms. *Computers and Electrical Engineering*, 67:672–689, 2018. doi:10.1016/j.compeleceng.2017.10.022.

[20] H. Meikle. *Modern RADAR Systems*. Artech House, 2008. doi:10.1109/PTGMTT.1963.1123258.

[21] Jakub Sochor, Roman Juránek, e Adam Herout. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*, 161:87–98, 2017. arXiv:1702.06451, doi:10.1016/j.cviu.2017.05.015.

[22] Inhwan Han. Car speed estimation based on cross-ratio using video data of car-mounted camera (black box). *Forensic Science International*, 269:89–96, 2016. URL: http://dx.doi.org/10.1016/j.forsciint.2016.11.014, doi:10.1016/j.forsciint.2016.11.014.

[23] Jinhui Lan, Jian Li, Guangda Hu, Bin Ran, e Ling Wang. Vehicle speed measurement based on gray constraint optical flow algorithm. *Optik*, 125(1):289–295, 2014. URL: http://dx.doi.org/10.1016/j.ijleo.2013.06.036, doi:10.1016/j.ijleo.2013.06.036.

[24] Peng Zhao. Parallel precise speed measurement for multiple moving objects. *Optik*, 122(22):2011–2015, 2011. URL: http://dx.doi.org/10.1016/j.ijleo.2010.12.022, doi:10.1016/j.ijleo.2010.12.022.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, e Alexander C. Berg. SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:21–37, 2016. arXiv:1512.02325, doi:10.1007/978-3-319-46448-0-2.

[26] Alexander C. Berg. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu. SSD300: Single Shot MultiBox Detector. URL: https://github.com/weiliu89/caffe/tree/ssd.

[27] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, e Piotr Dollar. Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:2999–3007, 2017. arXiv:1708.02002, doi:10.1109/ICCV.2017.324.

[28] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, e Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[29] Tsung Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, e Serge Belongie. Feature pyramid networks for object detection. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:936–944, 2017. arXiv:1612.03144, doi:10.1109/CVPR.2017.106.

[30] Jian Sun Shaoqing Ren, Kaiming He, Ross Girshick. Python Faster R-CNN. URL: https://github.com/rbgirshick/py-faster-rcnn.

[31] Joseph Redmon e Ali Farhadi. YOLOv3: An Incremental Improvement. 2018. URL: http://arxiv.org/abs/1804.02767, arXiv:1804.02767, doi:10.1109/CVPR.2017.690.

[32] Joseph Redmon e Ali Farhadi. YOLO - You Only Look Once. URL: https://pjreddie.com/darknet/yolo/.

[33] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, e C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755, 2014. arXiv:1405.0312, doi:10.1007/978-3-319-10602-1-48.

[34] Satish Chandra e Anish Kumar Bharti. Speed Distribution Curves for Pedestrians During Walking and Crossing. *Procedia - Social and Behavioral Sciences*, 104:660–667, 2013. URL: http://linkinghub.elsevier.com/retrieve/pii/S1877042813045515, doi:10.1016/j.sbspro.2013.11.160.